На DVD Ubuntu 8.04: специальная LXF-редакция



плюс: **Mandriva One 2008 Free**

Дружественный дистрибутив

Compiz » Atomic Tanks » Inkscape » и так далее!

LINUX FORMAT

Главное в мире Linux

Привет сисадминам!
Новая рубрика – специально для вас с. 44

Июль 2008 № 7 (107)

Настольная революция!

Советы и хитрости по повышению эффективности вашей работы



41 страница учебников

Unix-культура
Разберемся в классике хакерства с. 40

20 суперпрограмм
Лучшие приложения, о которых вы не слыхали с. 28

Взломай свой Eee
Как получить больше, чем может предоставить этот малыш с. 58

Ядро изнутри
Узнайте, как оно работает, и приобретите новые навыки с. 72



«Я вдруг понял, что существует потребность в свободной Flash-технологии»

Из бара в Брюсселе Роб Савой говорит о Gnash с. 26

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882
Каталог «ПРЕССА РОССИИ» – подписной индекс 87974

LINUX FORMAT
Главное в мире Linux

Что мы делаем?

» Мы поддерживаем открытое сообщество, предоставляя источник информации и место для обмена мнениями.

» Мы помогаем нашим читателям узнать больше о Linux в нашей рубрике Учебники: здесь каждый найдет себе что-то по вкусу!

» Мы лицензируем весь код, который появляется на страницах журнала, на условиях GPLv3.

» Мы стараемся предоставлять максимально точные, непредвзятые и актуальные сведения о мире Linux.

Кто мы?

В этом номере мы сфокусировались на подстройке GUI. Поэтому вопрос к команде LXF был таким: «Какие изменения вы сделали, чтобы ваш рабочий стол стал более эффективным?



Грэм Моррисон
Полоски на рабочем столе и спойлер над панелью задач – классика!



Майк Сондерс
Я подправил утилиту, защищающую от туннельного синдрома. Теперь она пишет: «Кончай мечтать – давай, работай!»



Нейл Ботвик
Я пересобрал X, Gnome, KDE со всеми ключами GCC. Действует безупречно: всего два сбоя в час!



Эфраин Эрнандес-Мендоса
Звуки маракас при старте и девственно-белые обои помогают победить сезонные расстройства.



Эндрью Грегори
Звукосниматели, «капелька» – и вы играете, как бог! Эээ, постойте, а что это за журнал?



Энди Ченнел
Я устал жать на Alt+Tab, чтобы переключаться между окнами, и купил шесть 22-дюймовых LCD-мониторов!



Дэвид Картрайт
Главное – громкие слова! Синергируйте и раскройте потенциал вашего GUI... или просто возьмите IceWM.



Майкл Дж Хэммел
Как художник, я всегда держу на рабочем столе парочку пикселей... На случай, если они закончатся в мониторе.



Джон Брэндон
Разве XML и Web 2.0 – не средство от всех болезней? Я работаю над оконным менеджером для социальных сетей.



Д-р Крис Браун
Пару недель назад я заменил старую механическую мышь новой оптической. А что, уже 2008 год?



Ник Вейч
Никогда не поддавайтесь на уговоры Майка дать ему разогнать ваш ПК! Спустя три недели я все еще правил xorg.conf в Vim. Ничего, Сондерс, придет и твой час...



Энди Хадсон
Я прогнал диски с дистрибутивом через аэродинамическую трубу, и теперь рабочий стол просто летает!



И снова о спирали

» В середине июля я внезапно слег с гриппом, что само по себе случается довольно редко. Не подумайте, что я собрался жаловаться, стонать и сетовать на судьбу-злодейку. Просто внезапно появившееся свободное время, которое подарили мне наши маленькие друзья-вирусы, натолкнуло меня на занятные размышления, которыми я хотел бы с вами поделиться.

Часто говорят, что Free Software – это модель наработки научных знаний, примененная к созданию ПО: здесь-де то же рецензирование, общедоступность и так далее. Немногие догадываются, что наука была такой не всегда. Так, до эпохи Ньютона основным критерием истинности была ссылка на классика: Плутарха, Аристотеля. Знакомые с математикой наверняка слышали о формуле Кардано, позволяющей решать кубические уравнения. А известно ли вам, что ее публикация привела в негодование математика и инженера Никколо Тарталью, который обвинил Кардано, говоря современным языком, в нарушении NDA? Причина (по мнению некоторых) проста – с помощью этой формулы Тарталья рассчитывал конструкции и удерживал монопольное положение на рынке.

Что же происходит сейчас? От коммерческих тайн и секретов мы плавно переходим к открытым стандартам и технологиям. Быть может, и проприетарное ПО – лишь естественный этап в развитии отрасли? **LXF**

Валентин Синицын » Главный редактор info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru
› **Адрес редакции:** Россия, Санкт-Петербург, Лиговский пр., 50, стр. 15.
› **Телефон редакции:** +7 (812) 640-49-90. Дополнительная информация на стр.128

LINUX FORMAT
Июль 2008 № 7 (107)

Содержание

Весь номер – прямо как на ладони: приятного чтения!

Обзоры

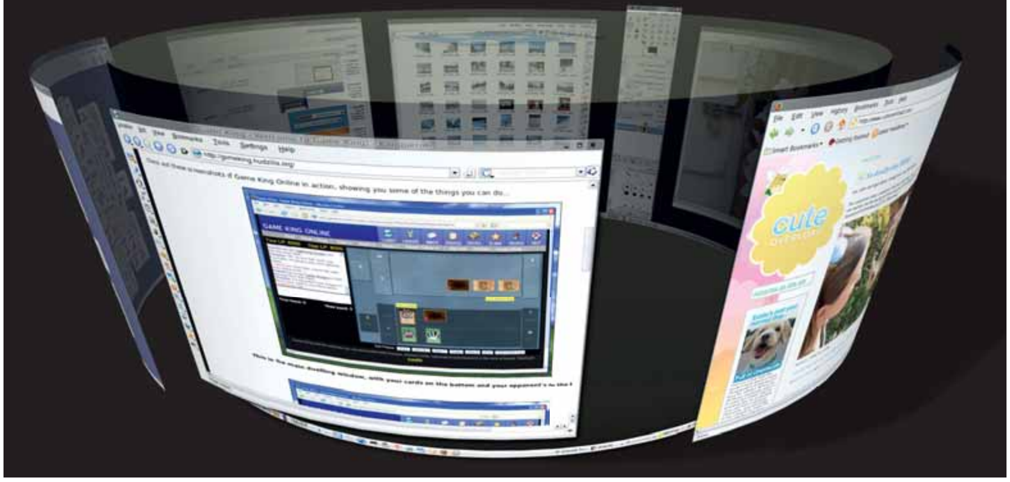


› **Наконец-то: что-то помощнее и попроще, чем MythTV.**



Настольная революция

Ваше окружение, ваши правила с. 32

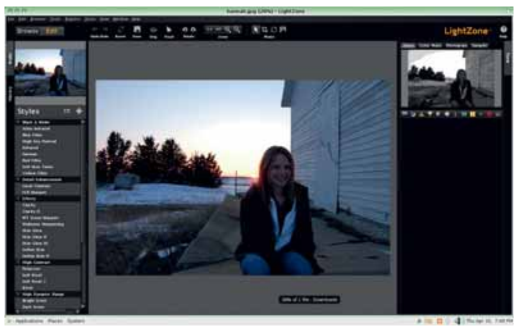


Что за штука... Akonadi

KDE, PIM и прочие ТБС: что все это значит? с. 48



20 незаменимых приложений с. 28



Люди говорят

«Вообще-то я интересуюсь встраиваемыми системами, когда-то даже делал первый Linux-порт GCC на 68K.»

Роб Савой объясняет, как он взялся за Gnash с. 26



В помощь новичку



Изучайте Linux по нашим учебникам!

Mandriva 2008 Spring Free

Linux для каждого: мощный и простой в использовании дистрибутив.

» Дружественный инсталлятор » Программы для Web и офиса » Мощные инструменты администратора

Спецрепортаж



Подпишись на Linux Format и сэкономь!



Постоянные рубрики



Учебники

Новости

ГЛАВНЫЕ НОВОСТИ: » Oyster выбирает Linux: за и против » ODF, очевидно, победил? » Птица-бегун – самая быстрая в мире » Лица Free Software » Firefox 3 и Opera 9.5 » Linux в Библиотеке Академии Наук

» Рубрику ведет Илья Шпаньков



Зачем устрице свобода?

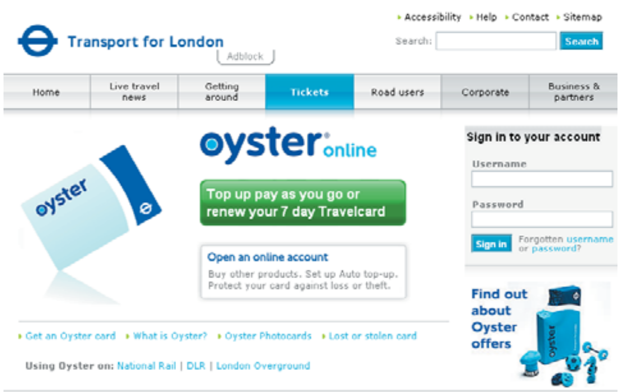
Как правило, мы всегда с радостью воспринимаем новости о переводе того или иного предприятия или проекта на рельсы Open Source. Не стало исключением и сообщение о том, что система бесконтактных платежей за пользование метрополитеном и другим транспортом Лондона под названием Oyster (в переводе на русский – «Устрица») в конце мая закончила частичный переход на свободное программное обеспечение. В частности, в целях повышения производительности и надежности этой необычной разработки большая часть используемого ранее ПО была заменена на Red Hat Enterprise Linux, web-сервер Apache и JBoss, при этом расходы на лицензирование программного обеспечения снизились почти на 80 %.

И все бы ничего, но в скором времени эта, на первый взгляд, ощутимая победа над «проприетарщиной» подверглась основательной критике с самой неожиданной стороны. Шутка ли – сам отец-основатель Free Software Ричард Столлмен [Richard Stallman] призвал лондонцев не пользоваться данной системой, а по старинке платить за проезд наличными. Не секрет, что высказывания патриарха движения за свободное ПО нередко критикуются противниками за излишнюю идеологизацию, но в этот раз даже убежденные сторонники принципов Free Software погрузились в сомнения: не перегибает ли палку их наставник и идейный вдохновитель?

Камнем преткновения стал тот факт, что помимо возможности осуществления платежей, система Oyster сохраняет в единой базе данных все записи о том, на какой станции метро гражданин вошел и на какой решил сменить подземную электричку на пешую прогулку по улицам Лондона, а также (уже в целях идентификации каждого индивидуума) оперирует и другими личными данными. Именно этот факт и заставил Столлмана высказать свое негативное отношение к данному проекту. По его словам, ирония судьбы заключается в том, что свободное программное обеспечение при использовании в системе Oyster будет работать как ограничитель свободы личности, позволяя госслужбам отслеживать и фиксировать все перемещения гражданина, причем без его ведома.

Конечно, новые удобные технологии являются естественным результатом научно-технического прогресса, но не будем забывать и о том, что нередко самые прогрессивные разработки оборачивались против их создателя – человека, когда начинали использоваться в работе военных или спецслужб. И в этом ракурсе можно понять Столлмена, который всю свою жизнь посвятил приучению окружающих к свободе и вдруг увидел, что его детище начинает использоваться в качестве ограничителя этих самых свобод. В конце концов, высказывания RMS против DRM и «тивоизации» тоже поначалу воспринимались с улыбкой, но впоследствии оказалось, что старик, черт возьми, был-таки прав. Поэтому имеет смысл более вдумчиво подойти к «проблеме лондонских устриц» и в будущем, возможно, попытаться исключить вероятность использования свободного ПО в аналогичных проектах, дабы не стать случайно создателем средства контроля и управления самим собой со стороны совершенно неизвестных вам личностей.



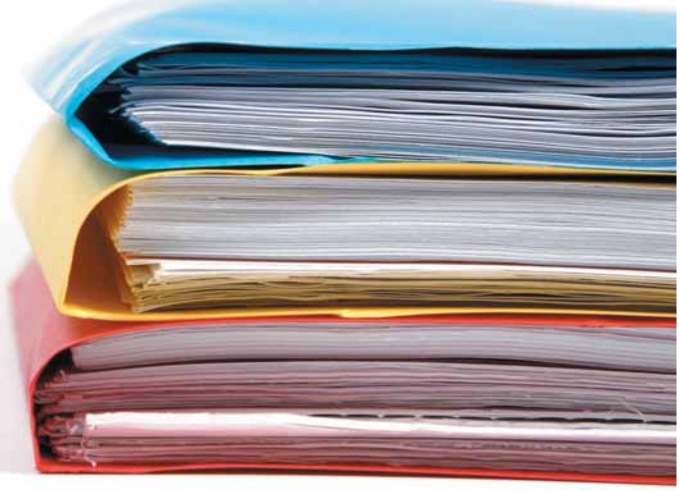


Новости короткой строкой

- Вышел первый релиз платформы Moonlight, реализующий технологию Microsoft Silverlight в Linux.
- Разработчики проекта Firebird объявили, что новые версии открытой СУБД SQL Firebird войдут в состав дистрибутивов Debian experimental, OpenSUSE 11 RC1 и Ubuntu 8.10 Intrepid Ibex.
- Компания ALTLinux объявила об электронном выпуске серверного дистрибутива ALT Linux 4.0 Office Server и о начале сертификации по программе ALT Linux Certified Specialist.
- Вышла новая версия офисного пакета OpenOffice.org 2.4.1, а компания Инфра-Ресурс подготовила адаптированный для российских пользователей вариант OpenOffice.org 2.4.1 Pro.
- Компания ASPLinux выпустила новую версию ASPLinux Server V Update 2.
- Состоялся финальный релиз дистрибутива OpenSUSE 11.0.
- Министерство обороны США включило SUSE Linux Enterprise Server 10 SP2 в «Список рекомендованных продуктов».
- Разработанные на основе ALT Linux дистрибутивы Легкий Линукс, Линукс Юниор, Линукс Мастер и Линукс Терминал прошли приемку и тестирование и включены в состав финального тиража ПСПО для российских школ.



Победа откладывается



Фуршеты закончились, шампанское выпито, премии получены и потрачены – казалось бы, все хорошо и можно слегка расслабиться. Но именно в такие моменты вдруг выясняется, что на самом деле ничего не закончилось и более того – результат может оказаться просто нулевым. В подобной ситуации сейчас оказались менеджеры компании Microsoft, отвечающие за продвижение формата документов OOXML в качестве международного стандарта. В начале текущего года комитет по стандартизации ISO большинством голосов принял положительное решение по данному вопросу и даже начал уже публиковать многостраничные спецификации свежеиспеченного стандарта, но тут произошло то, чего уже никто не ожидал: пять стран, участвовавших в голосовании по OOXML подали в ISO официальные протесты против ускоренной процедуры принятия нового стандарта. Первыми указали на поспешность решения по ISO представители ЮАР, к ним подключилась Бразилия, а пару дней спустя аналогичные протесты поступили от Венесуэлы, Индии и Дании. Основных поводов для подачи жалобы было два: отсутствие единого решения по внесению изменений в спецификации OOXML и несостоявшееся индивидуальное обсуждение замечаний. Дания же просто указала на неточность в регламентных документах, где сроки подачи апелляции оказались различными. Как бы то ни было, но подобное массированное выступление оказало свой эффект: публикация спецификаций нового стандарта ISO/IEC DIS 29500 была приостановлена до конца июня – по правилам международного комитета ISO жалобы должны быть рассмотрены в 30-дневный срок.

На самом деле, вряд ли подобные шаги заставят ISO отменить решение о принятии нового стандарта. Скорее всего, будут внесены некие поправки, проведены формальные заседания и выполнены другие малозначимые мероприятия, призванные, в основном, удовлетворить требования авторов апелляций. Слишком много сил и средств потратила корпорация Microsoft на это важнейшее для себя мероприятие. Не будем забывать, что принятие OOXML в качестве стандарта открывает для *MS Office* широкую дорогу в различные государственные и коммерческие структуры. Учитывая же, что львиную долю прибыли корпорация получает от продаж именно этого ПО, нетрудно догадаться, что все силы будут приложены к тому, чтобы возобновить процесс публикации спецификаций. Впрочем, поживем – увидим.

» Пока верстался номер

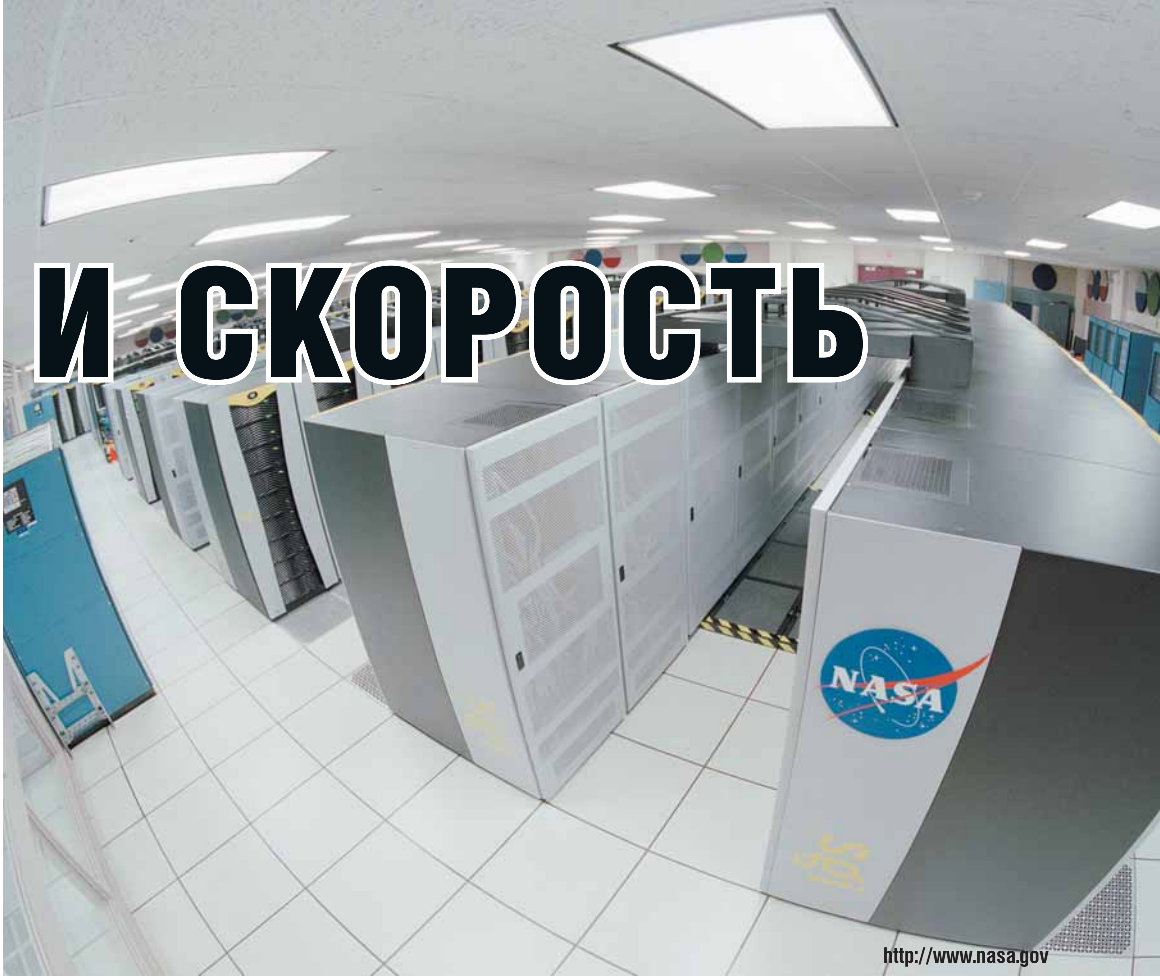
«ODF, очевидно, победил», – таковы слова технического директора Microsoft USA Стюарта МакКи [Stuart McKee], произнесенные им в ходе общей дискуссии на Red Hat Summit в Бостоне. Поддержка OpenDocument в *MS Office* ожидается с выходом версии 2007 SP2.

Компании Mandriva 10 лет

За чередой релизов и появлением новых программных продуктов незаметно подошла круглая дата – компании Mandriva и одноименному дистрибутиву исполнилось десять лет. Появившийся на свет в 1998 году французский проект по созданию дистрибутива GNU/Linux и сопутствующая компания Mandrake Soft за прошедшие годы значительно возмужали и окрепли. Остались позади нелегкие финансовые дни, приведшие к банкротству компании, выход на международный рынок и объединение с бразильской фирмой Connectiva (в связи с чем и образовалось новое имя – Mandriva), появление филиалов в других странах (в том числе, и в России), выход на корпоративный рынок – да много всего интересного и важного произошло за прошедшее десятилетие, так что повод собраться и отметить столь важное событие более чем основательный. Поэтому в последние выходные мая в Париж съехались все, кто принимал и продолжает принимать активное участие в развитии одного из самых известных в мире дистрибутивов GNU/Linux. Вопрос о помещении даже не обсуждался: где, как не на Эйфелевой башне должен отмечать важное мероприятие настоящий француз! В уютном конференц-зале, над крышами Парижа состоялось веселое торжество, в котором приняли участие руководители и разработчики компании Mandriva, а также съехавшиеся со всего мира партнеры и представители иностранных филиалов.

Впрочем, началось все, естественно, с официальной части: краткая история компании, сегодняшние достижения и планы на будущее – обо всем этом удалось узнать присутствующим гостям. К слову сказать, в поздравительных выступлениях принял участие и директор компании GNU/LinuxCenter Павел Фролов, являющийся главным инициатором появления филиала компании Mandriva в России, который рассказал об успехах Mandriva.ru и под бурное одобрение зала подарил французским коллегам плюшевого Тукса, неплохо прижившегося на российских просторах. По окончании официальной части прошел фуршет, но хотя французы и умеют, и любят развлекаться, программисты остаются программистами даже на подобных вечеринках: в разных концах зала постоянно возникали небольшие группы, собравшиеся вокруг подвернувшегося под руку ноутбука и увлеченно обсуждали различные технические вопросы. Конечно, всех желающих поздравить компанию с юбилеем не смогли бы принять и все залы Парижа, но для пользователей Mandriva было организовано другое масштабное мероприятие – Mandriva Install Fest, прошедший по всему миру и, конечно, во многих городах России в июне. Все пришедшие на встречу получили в подарок не только дистрибутив Mandriva, но и заряд хорошего настроения, сгенерированный участниками того самого юбилейного вечера на Эйфелевой башне.

Мощь и скорость



http://www.nasa.gov

Наверное, мало кого оставляют равнодушным соревнования Формулы-1. Поневоле проникаешься уважением к этим неказистым с виду болидам, которые лишены всяческой бытовой мишуры, являя зрителям мощь и скорость в чистом виде – результат синтеза самых новейших автомобильных технологий. Особенно живой интерес к гонкам проявляют автолюбители: кто из них не мечтает о том, чтобы получить в свое распоряжение все эти лошадиные силы и разработки светлейших голов автоиндустрии. Нам, пользователям персональных компьютеров, проще: казалось бы, установлен обычный дистрибутив GNU/Linux, но на самом деле он же используется и в мощнейших суперкомпьютерах, способных выполнять триллионы операций в секунду. И, надо сказать, в данной области науки и техники у нашей любимой ОС мало достойных соперников, посему и состязаться приходится с самим собой.

Совсем недавно компания IBM объявила о завершении работ по созданию очередного супергиганта под названием Roadrunner. Он собран на базе комбинации из 6948 двухядерных процессоров AMD Opteron и 12960 Cell, а его производительность составляет немыслимую доселе цифру в 1 петафлоп. На сегодняшний день это – самый быстрый суперкомпьютер в мире, почти в два раза обогнавший своего предшественника – также детище IBM и также работающий под управлением GNU/Linux суперчислогрыз Blue Gene. Использоваться он будет в Национальной лаборатории в Лос-Аламосе для нужд Министерства обороны США, а также для исследований в области астрономии и генетики.

Японская же новинка T2K Open Supercomputer, разработанная совместно с Университетом Киото и Университетом Цукуба, не может соперничать со столь мощными гигантами (ее быстродействие составляет всего 140 трлн. операций в секунду), но этот богатырь вошел в десятку самых мощных и быстрых суперкомпьютеров в мире, заняв при этом первую строчку в самой Японии. В его задачи будет входить анализ генетического кода и моделирование сложных химических реакций. Работает он, как вы догадались, также под управлением GNU/Linux. Всего же около 75% всех суперкомпьютеров в мире предпочитают доверять свои недюжинные возможности операционным системам GNU/Linux, при этом около 40% из них используют SUSE Linux Enterprise Server.

Место в истории



О роли личности в истории можно спорить вечно, доказывая как неизбежность ее появления, так и невозможность построения именно таких событийных цепочек без некоей конкретной исторической фигуры. Впрочем, сложность данного вопроса совсем не умаляет заслуг реальных людей, которые вольно и или невольно уже стали не только участниками, но и ключевыми действующими лицами того или иного исторического события. Не является исключением и мир свободного ПО, причем, изначальный демократизм всего явления в целом позволил проявить свои способности множеству лучших представителей нашего общества. Популярное онлайновое издание Free Software Magazine недавно решило опубликовать список наиболее заслуженных людей, сделавших мир таким, какой он есть сейчас. Что ж, с данной подборкой лучших представителей движения Free Software, наверное, кое-где можно и поспорить, но это не мешает опубликовать ее на страницах нашего журнала.

» **Ричард Столлман** [Richard Stallman] – основатель проекта GNU и всего движения Free Software;

» **Памела Джонс** [Pamela Jones] – создатель и редактор сайта Groklaw, автор более тысячи статей, рассказывающих обществу о свободном ПО;

» **Линус Торвальдс** [Linus Torvards] – создатель и ведущий разработчик ядра Linux;

» **Марк Шаттлворт** [Mark Shuttleworth] – основатель проекта Ubuntu Linux;

» **Ларри Пейдж** [Larry Page] и **Сергей Брин** [Sergey Brin] – основатели компании Google;

» **Боб Янг** [Bob Young] и **Мэтью Сцулик** [Matthew Szulik] – основатель и первый руководитель компании Red Hat;

» **Джимми Уэльс** [Jimmy Wales] – создатель Wikipedia;

» **Лоуренс Лессиг** [Lawrence Lessig] – автор лицензий Creative Commons;

» **Тим Бернес-Ли** [Tim Berners-Lee] – создатель World Wide Web и автор спецификаций HTTP и HTML;

» **Блейк Росс** [Blake Ross] – руководитель проекта по созданию браузера Firefox;

» **Дрис Байтерт** [Dries Buytaert] – автор Drupal;

» **Кейт Паккард** [Keith Packard] – инициатор появления ветви разработки X.Org, отпочковавшейся от XFree86;

» **Брэм Коэн** [Bram Cohen] – основатель BitTorrent;

» **Майкл Тиманн** [Michael Tiemann] – основатель Cygnus, первой в мире компании, построившей бизнес исключительно на открытом ПО.

Время собирать релизы

Июнь ознаменовался сразу двумя важными событиями: выходом финальных версий браузеров *Firefox 3* и *Opera 9.5*. Обилие новшеств и значительное улучшение поддержки современных web-стандартов в данных популярных программах ставит под удар и без того заметно пошатнувшиеся позиции некогда монопольно распространенного *Internet Explorer*. Чем же порадуют нас разработчики браузеров на сей раз? Вот какие новшества получили в свое распоряжение пользователи:

Firefox 3

» Обновленный менеджер паролей, позволяющий осуществлять вход на защищенные страницы без дополнительных действий пользователя;

» Новые элементы в адресной строке, значительно ускоряющие и улучшающие процесс обработки и сохранения страниц в закладках;

» Встроенные системы защиты от фальшивых web-сайтов, от вредоносных программ, расширенные возможности родительского контроля;

» Полноценное масштабирование просматриваемых web-страниц;

» Новый встроенный менеджер дополнений, позволяющих быстро расширять функциональные возможности браузера;

» Улучшенные возможности поиска и управления уже посещенными web-страницами;

» Родные элементы управления на всех поддерживаемых платформах и обновленный дизайн пользовательского интерфейса (в частности, кнопки Вперед и Назад объединены в одну)

» Повышена скорость работы, устранены утечки памяти (теперь для ее выделения и очистки используется библиотека jemalloc).

Opera 9.5

» Opera Link – технология, позволяющая синхронизировать данные между любыми версиями браузера, установленными как на компьютерах, так и на мобильных устройствах;

» Расширенный поиск – теперь в обычной адресной строке браузера можно осуществлять поиск ключевого слова во всей истории посещенных когда-либо web-страниц;

» Кардинальное изменение внешнего вида браузера, позволившее сделать работу в интерфейсе программы более интуитивной;

» Повышенная безопасность – в новой версии *Opera* появилось сразу два встроенных средства защиты, позволяющих не только избегать фальшивых web-сайтов, но и предотвращать доступ на компьютер всевозможных вредоносных программ;

» Скорость – в новой версии значительно увеличена скорость обработки содержимого web-страниц;

Надо отметить, что оба браузера с нетерпением ожидались аудиторией пользователей, что подтвердилось той активностью, с которой осуществлялись загрузки новых версий с серверов разработчиков. Только за первые пять дней было загружено около пяти миллионов копий *Opera 9.5*, а в случае *Firefox 3* показатели еще более впечатляющие: около 8 миллионов загрузок в первые сутки и около 15 миллионов – за пять дней с момента релиза. Следует также отметить, что на сегодняшний день общая доля браузеров из разряда «не IE» значительно выросла и ощутимо потеснила позиции *Internet Explorer*, при этом если в мире доля конкурентов встроенного в Windows браузера в общей сложности составляет около 25 %, то в России эта цифра ощутимо больше: на данный момент бывший монополист довольствуется примерно 65 % рынка. Учитывая, что новые версии *Internet Explorer* заметно уступают по функциональности и защищенности альтернативным браузерам, можно прогнозировать дальнейшее продолжение роста популярности и *Opera*, и *Firefox*.

ОБЪЯТЬ необъятное



Каждый, кто хоть раз бывал в городских библиотечных хранилищах, понимает, насколько сложной должна быть работа по систематизации и учету всех этих сотен тысяч разномастных томов, дожидающихся своих читателей на полках стеллажей. А теперь представьте, насколько увеличится сложность задачи в условиях, к примеру, Библиотеки Академии Наук РФ. Десятилетиями накапливающаяся научная литература, миллионы книг, плюс неизбежные изменения процедуры учета в течение всего периода существования библиотеки – в результате мы получаем практически неразрешимую задачу. Впрочем, не будем дальше интриговать и скажем, что в случае как раз Библиотеки РАН все проблемы решены и в качестве спасителя бесценных знаний для последующих поколений ученых выступило именно свободное ПО.

15 мая был полностью и успешно завершен проект «Внедрение современных телекоммуникационных технологий в информационную инфраструктуру Библиотеки Российской академии наук». Одним из активных участников данной инициативы государственной важности выступила компания ОАО ЛИНУКС ИНК, на долю которой и выпала задача по созданию программного обеспечения базы данных библиографических записей. И здесь как нельзя лучше проявила себя уникальная гибкость и открытость исходных кодов свободного ПО. Взяв за основу платформу Plone (кстати, являющуюся основой для создания «электронного правительства» во многих европейских странах), разработчики смогли в установленные сроки внести все необходимые изменения, специфичные для академической библиотеки, привели разнородные каталоги к единому формату хранения, соединили их в общую базу данных и организовали в ней эффективный поиск информации по ключевым словам. В результате, участники проекта смогли обеспечить более быстрый доступ к документальным и электронным информационным ресурсам путем создания современной автоматизированной инфраструктуры. Для того, чтобы лучше оценить масштабы проведенной работы, достаточно сказать, что на сегодняшний день Библиотека РАН ежегодно обслуживает более 23 тысяч ученых и специалистов, а также осуществляет обмен имеющейся в хранилище литературой с 78 странами мира. LXF

» При подготовке данного выпуска новостей были использованы материалы сайтов DaniWeb Blogs, International Organization for Standardization, InfoWorld, Mandriva Club, LinuxDevices, Free Software Magazine, Opera, Mozilla и Linux Ink.

LXF Репортаж

LGM2008: взгляд

Разработчик векторного редактора *sK1* **Игорь Новиков** представляет то, что сам он называет «предвзятым обзором третьей конференции Libre Graphics Meeting»



› **Игорь Новиков может говорить об sK1 часами, но на LGM выделили только 30 минут.**

В начале мая этого года авторы и пользователи свободного графического ПО в третий раз встретились наконференции Libre Graphics Meeting. Напомним читателям, что в первый раз эта конференция проходила в Лионе (Франция) в 2006 году, а ее вторая версия состоялась в Монреале (Канада) в 2007. На этот раз местом встречи встречи стал Вроцлав (Польша). Для свободного графического ПО конференция LGM, говоря словами классиков, событие «архинужное и архиважное». Но в чем его важность? Попытаемся ответить на этот и другие вопросы вместе с Питером Линнелом [Peter Linnell] из проекта Scribus...

Linux Format: Что вы ожидали от этой встречи Libre Graphics Meeting? Оправдались ли эти ожидания? Были ли какие-нибудь сюрпризы?

Питер Линнел (ПЛ): Ну... Я ожидал, что мы сможем увидеть прогресс в областях, связанных с Cairo и управлением цветом. Очень понравились дискуссии по OpenICC (открытый стандарт управления цветом) – в Вроцлаве мы впервые встретились в очном порядке.

Игорь Новиков (ИН): Сказать по правде, зная распорядок конференции заранее, сюрпризов я не ожидал. Точнее, пытался приготовить сюрприз – анонс релиза *sK1*. Но увы, не все проблемы в коде были решены и выпуск пришлось отложить. Общее же впечатление от конференции – эта область открытого ПО и вовлеченные в нее инициативы бурно развиваются,что не может не радовать. Это крайне важно, так как многие проекты в той или иной степени связаны. Если наметится застой в каких-то приложениях или библиотеках, это прямо или косвенно отразиться и на других проектах.

LXF: А что хотелось бы отметить особо?

ПЛ: Наибольшее удовлетворение я получил от посещения самого Вроцлава. Прекрасный город и дружелюбные люди. По графическому ПО – это новые возможности, появившиеся в *Krita* и *Inkscape*. Во многих случаях это действительно инновации, а не копирование существующих функций проприетарного ПО.

ИН: Присоединяюсь! В Польше я был первый раз, и историческая родина (мои предки из Восточной Польши) мне очень понравилась. Бытует мнение, что поляки заносчивы – это неправда. Открытый и

с двух сторон

дружелюбный народ. И что приятно – зная русский и украинский, не испытываешь проблем в общении. Не хотел бы подчеркивать отдельные моменты в плане открытого: наиболее важным мне показался общий подъем в отношении свободной графики.

LXF: Какие тенденции вы видите в развитии свободного графического ПО?

ПЛ: Основная тенденция – увеличение числа зрелых графических приложений, которые уже пригодны для профессионального использования. Благодаря совершенствованию инструментариев, таких как *Gtk* и *Qt*, разработка программ упрощается. Ну и SVG будет продолжать укреплять свои позиции в качестве хорошо документированного, открытого стандарта для обмена векторной графикой.

ИН: Для меня тенденция заключается прежде всего в том, что открытые графические приложения начинают постепенно теснить проприетарное ПО. Конечно, это происходит медленно, но зато явно видно, что идея Open Source завоевывает все больше и больше сторонников. Например, на соревновании Hackontest (hackontest.org) лидирующие места занимают не серверные приложения, а типично дизайнерские – *Scribus*, *Gimp* и *sK1*.

LXF: Какие изменения вы ожидаете увидеть в ближайшем будущем?

ПЛ: Наиболее значимое на данный момент – это улучшение управления цветом в большем количестве приложений. Некоторые, например, *Scribus*, *Cinepaint*, *Digikam* и *Krita*, уже неплохо работают с цветовыми профилями. *Gimp* и *Inkscape*, тем не менее, отстают. Серьезной проблемой остаются закрытые драйвера для калибровочного оборудования. Правда, несмотря на отсутствие поддержки от производителей, понемногу выполняется их обратный инженерный анализ.

ИН: На мой взгляд, в ближайшем будущем стоит ожидать существенного количества прецедентов по переходу небольших полиграфических предприятий на свободное ПО. И, соответственно, разработчики доведут свои приложения до уровня, пригодного к использованию в повседневной практике. Лично для меня sK1 Project можно будет считать успешным, если пользователи начнут говорить, что они просто работают на нашем приложении – это будет высшая оценка.

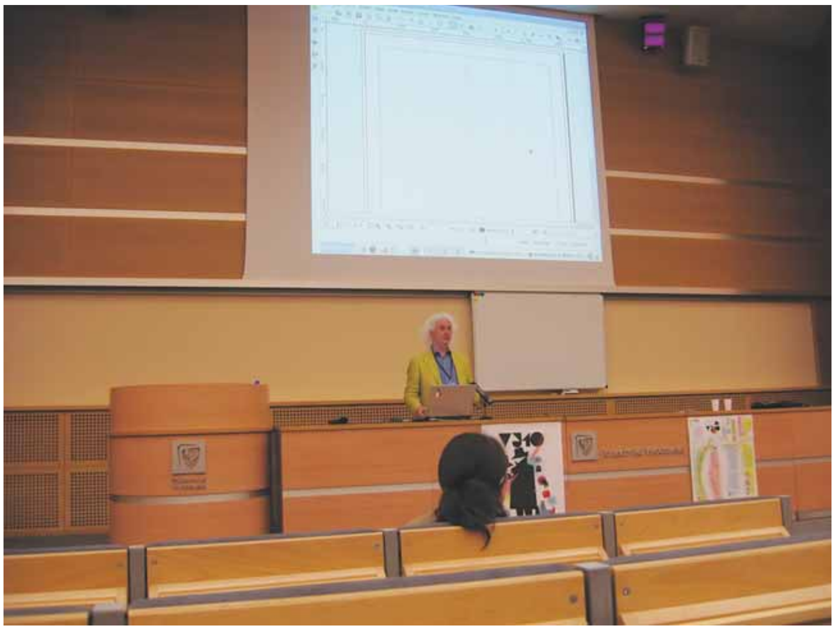
LXF: Что бы вы пожелали сообществу свободного ПО?

ПЛ: Хочется пожелать продолжения очень дружественного и продуктивного общения, которое сложилось на протяжении последних двух конференций. Я вижу реальную пользу в общности, единых целях и перспективах для графических приложений, которые сотрудничают. Это уже очень гармонично и взаимовыгодно. Я думаю, будущее продемонстрирует нам превосходство открытых стандартов над проприетарным ПО. Хороший пример – развитие формата OpenRaster.



ИН: Ну а я бы пожелал сообществу дальнейшего расширения. Есть масса совершенно незатронутых аспектов, которых явно не хватает для развития свободных графических программ. Любой, кто разбирается в предметной области, может принести пользу сообществу: или начав свой проект, или влившись в существующий. Чем шире будут применяться свободные приложения, тем меньше проблем будет у их пользователей. Wszystkiego dobrego! [всего доброго! – *польск.*] LXF

› Ломаете голову, что за загадочный термин «вычисления на ходу» зачастил на страницах LXF? Поясняем!



› Луис Дежардин [Louis Desjardins] рассказывает о Scribus.



LINUX FORMAT Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



Алексей Федорчук
Его слабости – mass storage, разметка диска и файловые системы.

Безальтернативность: всегда ли это плохо?

Linux всегда ценился его пользователями за свободу выбора. В области дистростроения это нашло свое выражение в том, что инсталляторы наиболее прогрессивных дистрибутивов предоставляли все большие возможности выбора пакетов, а сами пакеты становились все более дробными. Апогей этой линии развития – Debian, в котором пакет авторский может делиться и делиться на множество автономно устанавливаемых пакетов дистрибутивных. С учетом зависимостей, разумеется, но, при желании, исключительно «жестких».

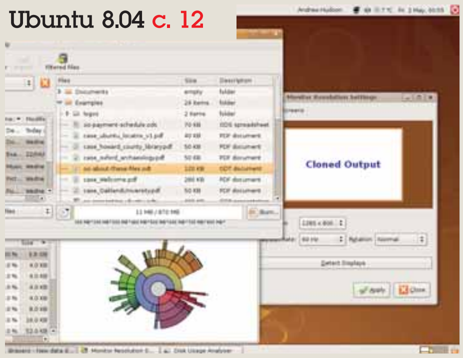
И вдруг – появление дистрибутивов с безальтернативной установкой строго заданного набора приложений. Наиболее ярко это проявилось в Ubuntu и Zenwalk. Несмотря на несхожесть, эти дистрибутивы позиционируются одинаково, предназначаясь для двух полярных групп пользователей: совсем начинающих и многоопытных, которые вдоволь наигрались в пересборку ядра, компиляцию «с нуля» собственной системы, индивидуальный выбор пакетов...

С точки зрения обеих групп, безальтернативность установки – это благо. Начинающий пользователь еще не знает назначения пакетов и не способен на сознательный выбор. А пользователь многоопытный выбирает не пакеты, а дистрибутивы, наиболее соответствующие его вкусам. А поскольку со временем абсолютная численность тех и других растет – увеличивается и популярность «безальтернативных» дистрибутивов.

alv@posix.ru

Сегодня мы рассматриваем...

Ubuntu 8.04 с. 12



› Кто такой: большой, коричневый и стоит на корпоративном сервере? Ну конечно Ubuntu 8.04!

Neuros OSD с. 17



› А кто такой: маленький, черненький и так и просящий: «Взломай меня!» Э-э-э-... это Neuros OSD!

НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT TOP STUFF

LINUX FORMAT Вердикт

Google Earth
Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» *Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.*

Рейтинг 9/10

Ubuntu 8.04 LTS

LINUX FORMAT На DVD

Хватит ли у этой цапли стойкости, чтобы выдержать суровый тест **LXF**? Разбирается **Энди Хадсон**.

Вкратце...

» Дистрибутив Linux с продленным циклом поддержки. См. также: CentOS или RHEL и SLED с договором на поддержку.

Ключевое ПО

» Gnome 2.22.1
» OpenOffice.org 2.4
» Firefox 3 beta 5
» Kernel 2.6.24
» GCC 4.2.3
» Pidgin 2.4.1

Любой наш читатель, кто не прожил последние четыре года в пещере, наверняка слышал об Ubuntu, одном из самых популярных ныне дистрибутивов Linux. И наверняка вы уже в курсе, что это вторая версия с долгосрочной поддержкой (Long Term Support, LTS) за последние четыре года, и от нее многого ждут. Теперь, получив готовый продукт, можно приглядеться поближе, копнуть поглубже и выяснить, что принес нам Ubuntu 8.04 LTS (он же – Hardy Heron, Стойкая Цапля).

Прежде всего следует отметить, что данная версия Ubuntu создавалась ради стабильности. В течение трех лет предполагаются обновления безопасности, но основные технологии никто не тронет. Для тех, кто «на работе», такая стабильность жизненно необходима: в конце концов, сколько крупных обновлений (не считая тьмы мелких патчей) претерпела Windows XP за семь лет с момента выпуска? «Домашний» же Linux-пользователь хочет постоянно быть в курсе последних событий, и версия LTS не для него.

Так что же интересного в дистрибутиве? Как мы уже упоминали, имеются Gnome 2.22.1 и ядро 2.6.24. Кроме того, *OpenOffice.org 2.4*, *Evolution* и *Rhythmbox* включены по умолчанию, плюс пара-тройка приложений сверх обычного набора. Однако, отступив немного назад, к процессу инсталляции, мы обнаружим важное дополнение. Это *Wubi*, Windows Ubuntu Installer, с которым пользователь может установить дистрибутив прямо из Windows, без переразбиения жесткого диска.



› **Hardy Heron взлетает – приземлится ли он на новых настольных ПК?**

> «С Wubi из Windows в Ubuntu можно перейти без переразбивки диска.»

Вуби-вуби-вуби-вуби!

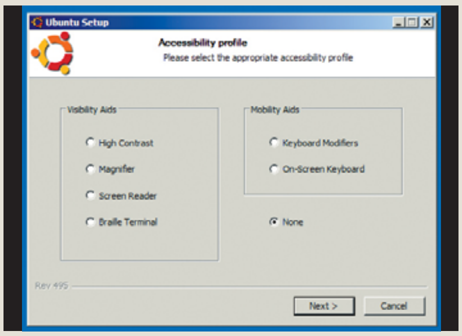
Вместо этого достаточно создать файл в Windows-системе, который *Wubi* использует для загрузки. Программа добавляет пункт в меню загрузчика Windows, что упрощает выбор системы при старте компьютера. Установка с помощью *Wubi* предельно проста: от пользователя требуется только ввести пароль и подтвердить его, и процесс пошел. После перезагрузки останется нанести несколько мелких штрихов (впрочем, автоматизированных) – и вы в Ubuntu-системе.

В этом Ubuntu безусловно преуспел: мостик для перехода из Windows – проще не бывает: даже диск разбивать не нужно. Весьма похвально, и это лишнее подтверждение и без того завидной репутации Ubuntu как системы, исключительно удобной в использовании.

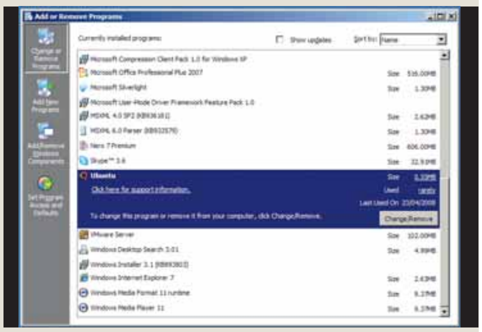
Шаг за шагом: Инсталляция с Wubi



1 Просто
Установка Ubuntu на Windows-машину с *Wubi* – сама простота! Просто следуйте инструкциям.



2 Доступно
Нужен доступ извне? *Wubi* поможет выбрать нужный вариант на этапе установки.



3 Чисто
Wubi так же запросто и удалит Ubuntu. Нечего бояться, уважаемый поклонник MS...

Но *Wubi* – только часть дистрибутива, хотя и важная. Для повседневной работы гораздо интереснее изменения в Gnome. Последняя версия Gnome, 2.22.1, лучшая из известных на сегодня, украсила Ubuntu отличными нововведениями. Первое и, возможно, самое полезное в наш век глобализации – отображение глобального времени на рабочем столе Gnome. Щелкнув на значке часов, видим обычный календарь; приглядевшись внимательнее, замечаем стрелочку выпадающего списка временных зон [Locations], где можно выбрать любые места, за временем в которых необходимо следить. А еще удобнее, что установить системное время и дату можно простым наведением курсора мыши на нужную зону с последующим щелчком.

Скажите «ч-и-и-з»!

Другое интересное приложение – *Cheese*, замена Apple Photo Booth для Gnome. По умолчанию *Cheese* не входит в Ubuntu, но программа доступна как пакет с полноценной поддержкой через *Synaptic*. Ubuntu легко распознал нашу камеру Logitech QuickCam Pro 3000, так что фото и видео мы снимали без проблем.

Как уже упоминалось, Ubuntu 8.04 предлагает долгосрочную поддержку. Однако, в отличие от Dapper Drake, это не касается Kubuntu, KDE-ответвления Ubuntu. Соответствующий анонс вызвал недоумение, но, если разобраться, причины такого решения вполне ясны. На момент выхода дистрибутива KDE 4 исполнилась всего пара месяцев – очень мало для включения в стабильный комплект. Успеет ли новый рабочий стол опериться? Вряд ли. Именно поэтому Canonical решила не включать его как часть стандартной инсталляции Kubuntu, оставив, тем не менее, KDE 3.5. Разработчики знают, что поддержка KDE 3.5 в будущие три года осложнится, так как активная разработка смещается на версию 4, и долго поддерживать устаревший рабочий стол не получится.



Грэм сказал

«У себя дома я недавно, первый раз в этом году, установил Mandriva. Это было подобно глотку свежего воздуха. Я люблю Ubuntu, но, чтобы сохранить динамику, нужно больше обновлений».

Вопрос Firefox

Hardy Heron претендует на стабильность и солидность. Дистрибутивом с одинаковым успехом могут пользоваться как индивидуальные пользователи, так и крупные корпорации. Тем более удивительно, что Canonical решила забежать вперед и включить в комплект бета-версию *Firefox 3* (Beta 5). Бесспорно, финальная версия *Firefox 3* не за горами [да, собственно, она уже вышла, так что и вопрос практически исчерпан, – *прим.ред.*].



› **Включая *Firefox 3*, Ubuntu испытывал судьбу – но сегодня бета-версию заменил полноценный релиз.**

Можно установить на Ubuntu и KDE 4, но это связано с загрузкой и инсталляцией такого количества разнообразных пакетов, что ради душевного спокойствия затевать это дело не рекомендуется.

Ныне Ubuntu дозрел до прямого вызова RHEL и SLED (по крайней мере, на рынке настольных ПК). Уже имея в запасе одну LTS-версию, дистрибутив намерен расширить область влияния в мире. Но – и это большое «но» – непохоже, чтобы Ubuntu вкладывал в разработку столько же сил, сколько вкладывают Red Hat и Novell накануне выхода своих генеральных версий.

> «Ubuntu дозрел до прямого вызова RHEL и SLED».

Слишком много, слишком скоро?

Hardy Heron держит тот же напористый 6-месячный цикл, что и все прочие версии Ubuntu (за исключением 6.06 LTS и 6.10, выходивших через девять и четыре месяца соответственно). Разработка RHEL 5 заняла 18 месяцев, позволив Red Hat стабилизировать платформу. Red Hat тесно взаимодействует с Fedora, постоянно отслеживая появляющиеся новшества, включая все ценное в RHEL и отбрасывая лишнее. Ubuntu этого лишен – каждая новая версия в основном базируется на предыдущей, и крупного риска Ubuntu до сих пор избегал. От Dapper (6.06) и до Hardy Heron включительно, проблем с дистрибутивом у нас не было. Вероятно, Ubuntu и Canonical просто боятся рисковать: если честно, их положение как нельзя лучше подходит для ломки стереотипов современных Linux-дистрибутивов.

Одних обновлений Gnome и косметических поправок к существующим приложениям в гонке за победу над Microsoft недостаточно. Да, Ubuntu применил PolicyKit, позволяющий дозированно выдавать права суперпользователя по мере необходимости. Да, есть PulseAudio и куча других технологий. Но ведь большинство других дистрибутивов или уже применяют все это, или вот-вот применят, наступая друг другу на пятки. Нам очень бы хотелось увидеть, как Ubuntu ломает стереотипы, использует новые методы ввода информации или новые технологии на рабочем столе. Ubuntu Hardy Heron – надежная платформа; почему бы не воспользоваться коротким циклом версий и в 8.10 отойти, наконец, от непрестанных обновлений и совершить что-нибудь подлинно революционное? LXF



› ***Cheese*, приложение Ubuntu для web-камер – ненамного более, чем безделушка, но позабавило нас на добрых пять минут.**

LINUX FORMAT Вердикт

Ubuntu 8.04 LTS

Разработчик: Canonical
Сайт: www.ubuntu.com
Цена: Бесплатно под GPL

Функциональность	7/10
Производительность	7/10
Простота использования	7/10
Документация	7/10

» *Ubuntu страдает застоем. Восхищаться особо нечем – хотя «деловым» людям наверняка понравится.*

Рейтинг 7/10

Axigen 6.0

Хотели было настроить себе почтовый Linux-сервер, да испугались загадочных команд и файлов конфигурации? **Крис Браун** пробует безболезненную альтернативу.

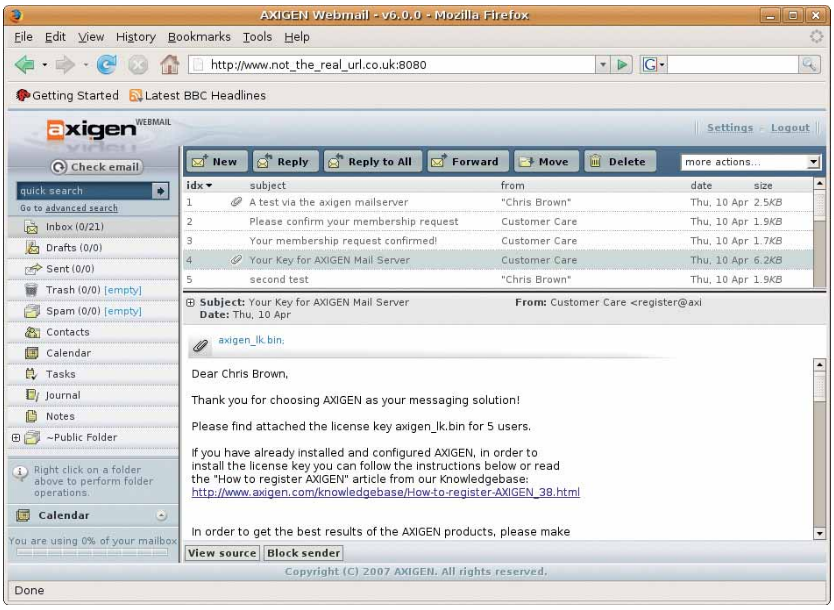
Вкратце...

» Почтовый сервер типа «все в одном» на Linux с простым графическим администрированием. Среди альтернатив открытого ПО – *Courier*.

Настройка почтового сервера всегда была чем-то вроде обряда посвящения в Linux-администраторы, а постоянные читатели наверняка помнят учебник по этой теме из LXF103. Тем интереснее было попробовать *Axigen*, главные козыри которого – простота установки и использования.

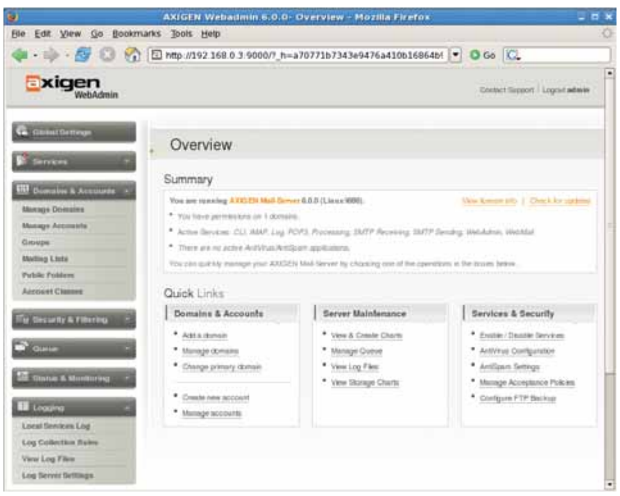
Axigen – это комплексный почтовый сервер. Здесь есть SMTP, POP и IMAP, web-сервер со «скинами», сервер списков рассылки, подключение к антивирусным и антиспамовым приложениям, а также календарь и дневник. В терминах свободного ПО это грубый эквивалент *Sendmail* или *Postfix* плюс *Dovecot*, плюс *SquirrelMail*. Продукт можно загрузить с сайта *Axigen*, как в формате RPM, так и в виде Deb-пакета. Все, что нужно сделать с полученным архивом – распаковать его и установить из root-терминала. Приложение появится в **/opt/axigen** – это один из немногих продуктов сторонней разработки, устанавливаемых именно туда, куда предписано стандартом FHS [Filesystem Hierarchy Standard].

> «Полная установка почтового сервера за 40 минут – неплохо».



» **У почтового сервера *Axigen 6.0* полноценный web-интерфейс. В данном случае прослушивается порт 8080, для совместимости с *Apache*.**

Базовую настройку поможет выполнить мастер. К примеру, нужно выбрать пароль администратора, указать «первичный домен», который будет обслуживать *Axigen* и определить запускаемые службы и прослушиваемые интерфейсы и порты. Это самая запутанная часть процесса, к тому же с уродливым интерфейсом в стиле *Ncurses*.



» **У почтового сервера *Axigen 6.0* полноценный web-интерфейс. В данном случае прослушивается порт 8080, для совместимости с *Apache*.**

Завершив ее, можно запускать комплекс. Все остальное настраивается через web-интерфейс, прослушивающий порт 9000. Из главного окна администратора (см. рис. внизу слева) можно добавлять новые домены, создавать в них почтовые учетные записи, управлять списками рассылки, проверять очереди писем и файлы протоколов, настраивать антиспамовые фильтры и делать много чего еще.

Раз, и готово!

Axigen имеет Flash-демо для помощи в первичной настройке. Продукт не претенциозен, но со своей задачей справляется. Мы взялись за дело и установили комплекс на существующий почтовый сервер (не забыв при этом отключить штатные SMTP и POP). 15 минут грузили, устанавливали и делали первичную настройку. Еще 10 минут ушло на загрузку лицензии, создание доменов и добавление новых пользователей. Честно говоря, были и еще 15 минут, пока мы вспоминали, как открываются дополнительные порты в брандмауэре. Пусть так, но 40 минут на полную установку почтового сервера – это совсем неплохо.

Axigen не бесплатен. Лицензия начального уровня стоит €250 и рассчитана на 25 почтовых ящиков. С этой точки цены растут до €2700 (редакция Service Provider Edition на 5000 ящиков). Для пробы можно загрузить демо-версию – она работает 30 дней безо всякой лицензии (но помечает каждое сообщение маркером об использовании пробного ПО). Можно также получить бесплатный ключ, действительный в течение года, но тогда программа сможет работать только с пятью ящиками, и только в пределах одного домена.

Истые Linux-хакеры обольют презрением мысль выложить €250 за то, что и так возможно с помощью свободных программ. Но для простых смертных это деньги, потраченные не зря. LXF

LINUX FORMAT Вердикт

Axigen 6.0

Разработчик: Gecad Technologies
Сайт: www.axigen.com
Цена: €250 (начальный уровень на 25 ящиков)

Функциональность	9/10
Производительность	8/10
Простота использования	9/10
Оправданность цены	6/10

» *Попробуйте, если вам нужен полноценный, но простой в использовании почтовый сервер.*

Рейтинг 8/10

CrossOver Games 7.0

CrossOver от Codeweavers – уже не скучная запускалка для офисных приложений! **Грэм Моррисон** рассматривает первую версию, созданную специально для компьютерных игр.

Вкратце...

» В Windows-игры можно играть, не покупая Windows. Альтернатива – *Cedega* и ручная настройка *Wine*.

Наша тестовая машина

» AMD Athlon 64 X2 5000+

» 1Гб ОЗУ

» Nvidia 7600GS 256Мб ОЗУ (PCIe)

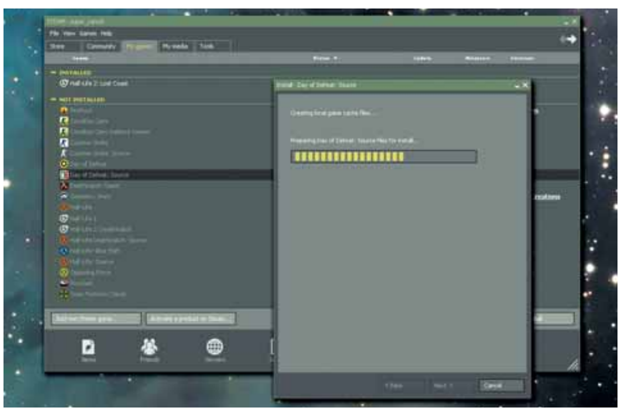
CrossOver – прямой соперник эмулятору *Cedega* от TransGaming, который мы обозревали в LXF94. Обе программы призваны помочь Linux-игрокам в получении новейших Windows-игр, и обе базируются на *Wine*, слое эмуляции Windows, который на самом деле не эмулятор.

Игры – это отход от основной деятельности *CrossOver*: до сих пор компания занималась обеспечением работы основных Windows-приложений (*Microsoft Office*, *Adobe Photoshop* и пр.) в Linux. Новейшие версии предлагают подобные функции даже для OS X от Apple. Однако, выход *Office 7* изрядно затянулся: Джереми Уайт [Jeremy White] – исполнительный директор CodeWeavers – намерен выпустить *CrossOver Linux 7*, только обеспечив работу в нем *Microsoft Office 2007*. Тем временем разработчики *Wine* достигли изрядного прогресса в игровой сфере, и в Codeweavers решили издать отдельную «развлекательную» версию *CrossOver*, не повязанную совместимостью с Office и сопутствующим процессом тестирования.



› *CrossOver Games* **непрерывно набирает силу и применяет новейшие версии Wine, добиваясь максимальной игровой совместимости.**

Двоичные пакеты доступны для Red Hat, Mandriva, OpenSUSE, Debian и Ubuntu, причем для Debian и Ubuntu есть 64-битные версии. Есть также пакет общего назначения, пригодный и для прочих дистрибутивов; процесс установки несложен. *CrossOver Games* работает в демо-режиме, предлагая семь дней полной функциональности, и только потом требует приобретения лицензии. Неплохой способ проверить, что получишь любимые игры, прежде чем выкладывать денежки! В стиле *CrossOver Linux*, инсталляция Windows-ПО происходит через окно настройки. Перед запуском установщика необходимо выбрать в списке игру, которую вы собираетесь добавить. Работая с онлайн-игровой системой Steam, *CrossOver* автоматически загружает необходимые компоненты Windows и Steam, запускает панель настройки Steam и добавляет все установленные Steam-игры в свое меню *CrossOver Games*.



› **Игры Steam (особенно от компании Valve) – козырь *CrossOver Games*.**

Вопрос выбора

Серьезная проблема *CrossOver* – малое количестве поддерживаемых игр. На момент написания статьи только 26 игр были отмечены знаком 'Silver' (серебряный статус) в официальной таблице совместимости. У *Cedega* их на несколько десятков больше. Если *CrossOver* помечает *Grand Theft Auto 2* (год выпуска 1999) как пригодную для запуска, то *Cedega* дает подобный статус *Grand Theft Auto 3* образца 2004. Кроме того, *Cedega* работает с *Oblivion* и *Battlefield 2142*. Но у *CrossOver* есть серьезное игровое преимущество: это Steam-игры от Valve. Работают и фантастическая *Team Fortress 2*, и культовая *Half Life 2: Episodes 1 & 2*, только приобретать нужно загружаемые версии, а не диски в местном магазине.

Игры Steam органично встраиваются в рабочий стол, и мы даже обнаружили, что они опережают по скорости *Cedega*. На нашей тестовой машине мы насчитали в среднем 36 кадров в секунду, играя в *Day of Defeat Source* при разрешении 1024x768 с включенным HDR, тогда как в *Cedega* больше 26 набрать не удалось. Жаль, что в *CrossOver* не действует сглаживание, что сказывается на качестве графики (над этим сейчас работают).

Можно подумать, что выбор между *Cedega* и *CrossOver* означает всего лишь выбор платформы для любимых игр. Все так, но для большинства Linux-пользователей к этому добавляется еще одно измерение. Codeweavers активно участвует в разработке *Wine*, передавая проекту многие из произведенных изменений; *TransGaming* этого не делает. В итоге, ваше решение определяется не только качеством ПО, но и этикой – голосуйте кошельком! LXF

LINUX FORMAT Вердикт

CrossOver Games

Разработчик: Codeweavers
Сайт: www.codeweavers.com
Цена: $39,95

Функциональность	8/10
Производительность	7/10
Простота использования	9/10
Оправданность цены	9/10

» *Коль скоро ваши любимые игры поддерживаются – настоятельно рекомендуем.*

Рейтинг 8/10



Mandriva 2008.1

Станет ли 2008 годом возврата былой славы Mandriva как самого дружелюбного из Linux-дистрибутивов? **Ник Вейч** вникает в детали…

Вкратце...

» Дистрибутив, известный своим дружелюбием к пользователю. Монополия постепенно нарушается наступлением Ubuntu и PCLinuxOS со товарищи.

Вернемся немного назад: 2007, вероятно, был самым неудачным годом для Mandriva. Ее почти затмили Ubuntu и перекроенная Fedora, и даже выскочки вроде PCLinuxOS (который и сам-то родом из Mandriva) снискали больше внимания и похвал. Mandriva же, некогда светоч передовых технологий, барахталась среди устаревших ядер, недоделанного ПО и прочих пережитков.

Ныне дистрибутив резко набирает форму. Весенняя версия обладает таким набором новейших открытых программ, какой только способен уместиться на одном DVD. Ядро 2.6.24.4 (ему всего-то несколько недель!) дополняется *Firefox 2.0.0.13*, Gnome 2.22.0, *Glibc 2.7*, *Gimp 2.4.5*, *OpenOffice.org 2.4*, последними проприетарными драйверами ATI и Nvidia и KDE 4 (полная и минимальная версии), а также 3.5.9.

Набор не ограничивается старыми знакомыми: есть и новинки. *Avant Window Navigator*, клон OS X Dock, выглядит блестяще и действует отменно. *Miro*, метаменеджер потокового медиа, запросто превратит Linux-машину в телевизор, непрерывно транслируя видео из различных источников.

Одно из более полезных, но менее заметных дополнений – *Conduit*. Это Gnome-инструмент, чья основная задача – легко и надежно синхронизировать данные между приложениями и/или устройствами хранения.



» Все модули расширения уже установлены – можно рассчитывать на «зеленый коридор» в Интернет.

Игры на любой вкус

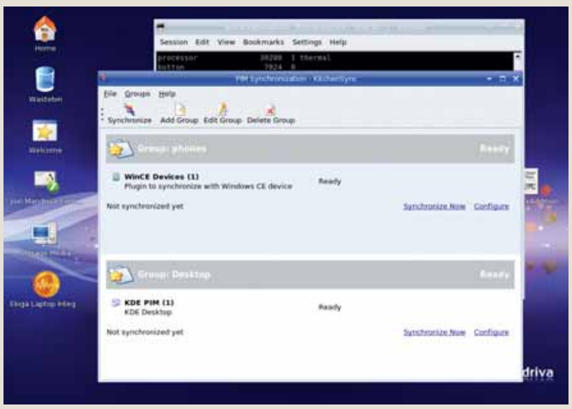
Версия PowerPack включает программу *Cedega* на основе *Wine*, позволяющую установить немало Windows-игр (хотя и не все). Если интересующая вас игра старше трех месяцев и не имеет какой-нибудь параноидальной защиты, все должно быть в порядке. Как минимум, на сайте *Cedega* приводится внушительный список поддерживаемых игр.

PowerPack обладает также комплектом кодеков для всевозможных аудио- и видеопрограмм – можно вызывать *Firefox* и двигать прямо на YouTube, забыв об установке модулей расширения и бесчисленных перезагрузках. Круто!

Взявшись за Mandriva впервые с прошлого года, с ходу изменений и не заметишь. На самом деле, изменилось все. Год назад Mandriva была устаревшим, неуклюжим и хромающим дистрибутивом. Сейчас она источает заботу и внимание к деталям, а если какая-то штучка и не работает – что ж, такова плата за прогресс.

Версия PowerPack интегрировала все доступные проприетарные технологии, хоть чуть-чуть способные расширить функциональность дистрибутива и сделать его удобнее (три месяца бесплатной поддержки и прочих услуг – это само собой). Если вы оказались в числе тех многих, кто пренебрегал Mandriva в прошлом году, проверьте: не пора ли вернуть на свой рабочий стол эту приветливую желтую звездочку. LXF

В унисон



» *KitchenSync* поддерживает все в актуальном состоянии.

Mandriva 2008.1 во всеуслышание заявляет о синхронизации с устройствами Nokia, BlackBerry, Windows Mobile. Охотно верим, видео смотрели: www.youtube.com/watch?v=ZMNpl3RUwo8. Увы, у нас не получилось. Mandriva корректно распознал наш Windows Mobile как RNDIS-устройство, но наотрез отказался с ним работать. Итак, придется идти путем проб и ошибок: синхронизация хороша, но не безупречна.

LINUX FORMAT Вердикт

Mandriva 2008 Spring PowerPack

Разработчик: Mandriva
Сайт: www.mandriva.ru
Цена: от 1475 руб.

Функциональность	9/10
Производительность	8/10
Простота использования	10/10
Оправданность цены	8/10

» *Может быть, эта версия изменит ваше мнение о Mandriva. Попробуйте прямо сейчас!*

Рейтинг 9/10

Neuros OSD

Если вы ищете домашний медиа-плейер с Linux, но не MythTV – посмотрите, что нашел **Грэм Моррисон**.

Вкратце...

» Подключаемый к телевизору медиацентр, способный записывать и воспроизводить видео, музыку, показывать фотоколлекции и т.д. Единственная альтернатива – *MythTV*, да и то требующая дополнительной аппаратуры для воспроизведения.

Из сотен управляемых Linux устройств, присутствующих на рынке, очень немногие имеют надпись ‘Powered by Linux’ на упаковке, и уж совсем единицы поощряют пользователей вторгаться в недра операционной системы. Так что Neuros OSD – приятный сюрприз. Это видеомагнитофон XXI века. Подключив к устройству чуть ли не любой видеоисточник, получишь отформатированное видео, пригодное для воспроизведения на телевизоре или портативном видеоплейере.

Упаковку украшают два логотипа: один – знакомый пингвинчик, другой гордо гласит ‘open source’. Открываем коробку – изумление растет. По размерам Neuros значительно меньше, чем кажется на фото в журнале: всего-то 14x14 см.

Устройство подключается к сети по Ethernet или беспроводному мосту (встроенной Wi-Fi-карты нет); видео можно подвести через гнездо SVideo сбоку на корпусе либо через RCA-подключение. Вывод ограничен RCA, провода входят в комплект вместе со звуковыми штекерами для ввода и вывода аудио. В Европе прилагаются также переходники SCART. Единственный функциональный недостаток – неспособность работать с HDTV.

Порты и слоты имеются для всех мыслимых накопителей: USB, Memory Stick Duo и Duo Pro от Sony, карт CompactFlash, SD и MMC. Встроенной памяти – 32 МБ, и в ней хранится ОС BusyBox Linux версии 1.6.1, но последние модели поставляются с картой CompactFlash, расширяющей этот объем. На карте записаны графические файлы основанного на *Qt* пользовательского интерфейса.

> «Neuros OSD оживает, когда его подключают к телевизору».



› **Внешность бывает обманчива. Neuros чуть объемнее, чем футляр для CD, а также имеет гнезда и слоты почти для любых вариантов подключений.**

Включил – и наслаждайся

Чтобы пробудить Neuros к жизни, достаточно подключить его к телевизору. Пользовательский интерфейс основан на иерархическом меню, прилагаемый пульт дистанционного управления позволяет переключаться между планированием записи телепрограмм и воспроизведением медиа из любого источника (включая разделяемые ресурсы Samba и серверы UPnP). Прилагаемый IR Blaster позволяет настроить Neuros на «инфракрасное» переключение каналов в спутниковой антенне или кабельном распределителе.

Навигация проста, и мы нашли ее безупречной, без проблем просмотрев фильмы в форматах DivX, *QuickTime* и H.263 при разных разрешениях экрана. Мы слушали аудиофайлы форматов Ogg Vorbis, MP3 и FLAC, жалея лишь об отсутствии цифрового аудиовыхода. Neuros переварил все, что мы ему «скармливали».

И еще одна вкусность Neuros – она связана с ярлыком ‘open source’: за исключением пары проприетарных кодеков, исходный код проекта опубликован целиком. Живое сообщество единомышленников постоянно работает над новыми функциями – некоторые из них уже вошли в официальный дистрибутив. Мы установили web-интерфейс и FTP-сервер, а Google по итогам Summer of Code 2008 обещает тюнер интернет-радио Last.fm, плейер *MythTV* и web-браузер.

Недавний переход на *Qt* сравнительно упростил написание и распространение приложений, поэтому Neuros может стать превосходной забавой для Самоделкиных. Или для меломанов. LXF

Свойства навскидку



YouTube

Neuros – великолепное решение для полноэкранной высококачественной трансляции передач с YouTube; и браузер не нужен.



Командная строка

Хотите войти в командную строку с помощью телевизионного пульта управления? Пожалуйста! Или входите в Neuros по Telnet.

Linux Format Вердикт

Neuros OSD

Разработчик: Neuros Technology
Сайт: www.neurostechnology.com
Цена: $179,99

Функциональность	10/10
Производительность	8/10
Простота использования	7/10
Оправданность цены	9/10

» *Neuros достаточно дешев для импульсивной покупки, а технологических тонкостей в нем столько, что увлечение может длиться годами.*

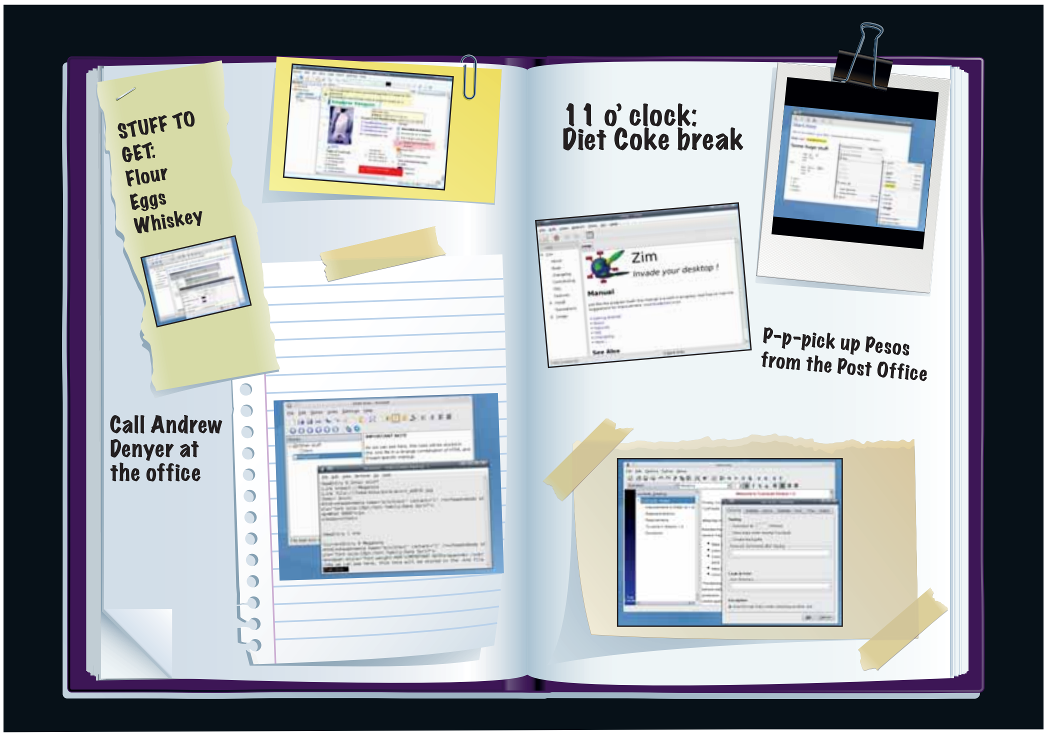
Рейтинг 9/10

Сравнение

» Каждый месяц мы сравниваем тысячи программ – а вы можете отдыхать!

Для заметок

Страдаете от информационной перегрузки? **Майк Сондерс** представляет шесть менеджеров заметок для борьбы с завалами мятых записок на столе…



Про наш тест…

Менеджеры заметок поневоле балансируют на грани: если функций у них слишком много, они громоздки и неуклюжи, если слишком мало – цена им едва ли выше, чем простому текстовому редактору. Разработчики обязаны следить, чтобы интерфейс не переполнялся всякими штучками-дрючками, чтобы мы, конечные пользователи, не чувствовали себя пилотами за панелью управления Boeing 747. Вот что мы решили выделить:

» **Интеграция с рабочим столом** Работает ли приложение в Gnome, KDE или *Xfce*? Можно ли перетащить в заметку текст или картинки?

» **Импорт/Экспорт** Можно ли использовать данные из другого менеджера заметок? Можно ли сгенерировать HTML-версию своих заметок?

» **Шифрование** Если в заметках хранятся пароли и PIN-коды, данные не должны быть легко доступными. Программа должна шифровать данные при их сохранении на диске и запрашивать пароль для доступа к ним.

Наш выбор

Бумага… кто ж ее не ненавидит? Живем мы в «век информации», а обещанных безбумажных офисов ждать не три года, а десятилетия. Наши столы завалены записками, памятками и клочками чего попало – надо бы их срочно разобрать, да времени нет. Вы наверняка пробовали компьютеризовать свои заметки «лобовой атакой», подвесив на рабочий стол текстовый файл (или документ) для спешной записи номеров телефонов, напоминаний и другой информации.

Поначалу это помогает, но скоро файл непомерно разрастается. И даже если вы располагаете записи по категориям и регулярно удаляете устаревшую информацию, все равно тонуть вам в болоте данных, в которых невозможно разобраться. Конечно, это удобнее, чем выискивать информацию на самоклеющихся листочках, и, безусловно, бережет природу, но должно же быть более элегантное решение.

Запишите это

Программы управления заметками помогают представить информацию в ясном и доступном виде. Заметке обычно можно назначить одну или несколько категорий. Хорошие менеджеры заметок не привязаны к типу информации – в них можно записать что угодно, от адресов URL и предстоящих событий до расписания поездов и названия песни, только что прозвучавшей по радио. Одни возможности, такие как поиск и копирование текста, жизненно необходимы, другие – форматирование (жирный шрифт, курсив) – удобны для структурирования данных.

К счастью, в Линуксландии подобных программ хватает, и у нас есть выбор. Сообщество открытого ПО так и сыплет утилитами, помогающими организовать нашу жизнь – от тяжело груженых функциями и снабженными мощным импортом/экспортом данных до легковесных, немногим отличающихся от обычного блокнота. Далее мы рассмотрим шесть самых выдающихся менеджеров заметок и поможем подобрать подходящую программу для дома и/или офиса.

Tomboy

Флагман Mono с простотой Gnome.

Tomboy – одно из самых известных (наряду с *Banshee* и *F-Spot*) приложений, написанных на C#/Mono. Номер версии 0.10.0 наводит на мысль о незрелости, но по стабильности в целом *Tomboy* близок к 1.0. Правда, его производительность страдает от Mono: время запуска программы достаточно велико, так как она должна подключить немало динамических библиотек Mono. Тем, кто работает с другими программами Mono, бояться нечего: соответствующие библиотеки уже загружены в оперативную память. Но если Mono в вашей системе нет, то большая задержка при запуске (десять секунд на тестовом компьютере) сильно раздражает, особенно когда нужно быстро что-то записать (например, при разговоре по телефону).

При *GTK#*-интерфейсе, *Tomboy* лучше сочетается с Gnome и *Xfce*, хотя прекрасно работает и в KDE. Если на вашем рабочем столе или в оконном менеджере есть системный лоток, то при запуске *Tomboy* разместит в нем иконку, щелкнув на которой правой кнопкой мыши, можно получить доступ к диалогу настроек и онлайн-справке. Интерфейс программы пугающе аскетичен – здесь нет ни меню, ни панелей, ничего такого, где можно было бы кликнуть. Это и правда поначалу может сбить с толку: как сделать шрифт жирным? Как создать новую заметку?

Все в контексте

Ответы лежат в контекстном меню, вызываемом правым щелчком мыши, и в иконке в системном лотке. Это одна из проблем, характерных для многих приложений Gnome: мало возможностей для настройки интерфейса «под себя». Да, *Tomboy* прост и не свихнет вам мозги раскиданными где попало виджетами. Но добавить кнопок на панели инструментов не помешало бы: например, для выделения текста жирным вовсе лишнее лезть в меню. Ужасно неприятно, что в *Tomboy* нельзя настроить интерфейс на свой вкус, однако есть набор комбинаций клавиш, которые можно изучать.

Для форматирования текста в *Tomboy* предусмотрены выделение жирным, курсивом и подчеркиванием, а также моноширинный шрифт (подходит для фрагментов кода), четыре кегля и маркеры для списков. Форматирование неплохо сохраняется при экспорте в HTML, но здесь возникает другая проблема: экспортировать можно только отдельные заметки. Так, если основная заметка содержит ссылки на другие, то для генерации HTML-версии придется пройтись по всем заметкам и экспортировать их, а затем поправить ссылки в HTML. Это происходит несмотря на опцию 'Export linked notes' [Экспортировать связанные заметки] в диалоге экспорта и выглядит очень убого – конечно, версия *Tomboy* всего лишь 0.10.0, но экспорт в ней фактически бесполезен.

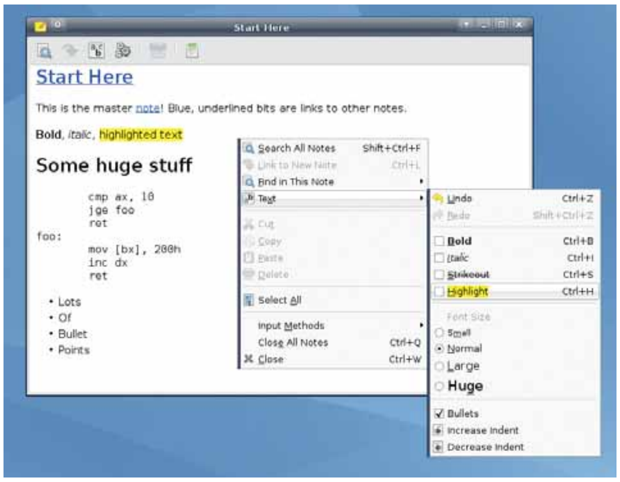
Неразумные имена

Сами заметки хранятся в XML-файлах с незабываемыми именами типа **10960f09-9977-428c-8f5a-6c3afb5df5a4.note**. Мы не против XML, но если понадобится прочитать заметку на другом компьютере и под рукой не окажется *Tomboy*, то копание в куче файлов со случайными именами вас не обрадует. Импорт заметок отсутствует, и если нужно перенести данные из другой программы, придется копировать и вставлять их вручную. *Tomboy* неплохо поддерживает перетаскивание текста, и когда вы делаете это из *Firefox*, базовое форматирование web-страницы сохраняется.

Как система категорий в программе используются «Блокноты» ('Notebooks'). Можно создавать новые блокноты, задавать им имена и затем связывать с ними отдельные заметки. При этом нельзя связать одну заметку с несколькими блокнотами. Невозможно зашифровать свои заметки или защитить их паролем. Из приятных мелочей:

> «Tomboy в целом стабилен, но кое-где есть недоработки.»

можно синхронизировать заметки с удаленным сервером через WebDav, настроить горячие клавиши для открытия заметок и включить wiki-подобный режим, так что, например, WordsWithoutSpaces превратится в «живую» ссылку для новой заметки.



Форматирование текста в *Tomboy* доступно через меню, но лучше бы предусмотреть на панели инструментов кнопки для основных операций (например, смены начертания и кегля).



Система модулей расширения упрощает добавление функций в *Tomboy*.

Возможности программы можно расширить с помощью модулей расширения ['Add-ins']; некоторые из них реализуют вышеупомянутый экспорт в HTML и синхронизацию по протоколу WebDav. Также есть модули для интеграции с Evolution – в заметку можно перетащить электронное сообщение, и в *Tomboy* появится ссылка с именем из темы письма – и импорта из Sticky Notes. Последнее, если верить документации, предполагает появление новой опции в меню Tools, чего на деле не происходит, и пользоваться этим нельзя.

Эти мелкие проблемы тормозят *Tomboy*; можно понять разработчиков, готовых урезать какие-то функции ради простоты использования (хотя побольше возможностей для импорта не помешали бы), но некоторые вещи сделаны наполовину или вообще не сделаны, и хотя приложение в целом стабильно, кое-где есть недоработки.

LINUX FORMAT Вердикт

Tomboy

Версия: 0.10.0
Сайт: www.gnome.org/projects/tomboy
Цена: Бесплатно по лицензии LGPL

» *Многообещающее средство управления заметками, но много теряет из-за неработающих функций и нехватки средств для настройки интерфейса «под себя».*

Рейтинг 6/10

KnowIt

Приложение KDE, достойное начинаться с K!

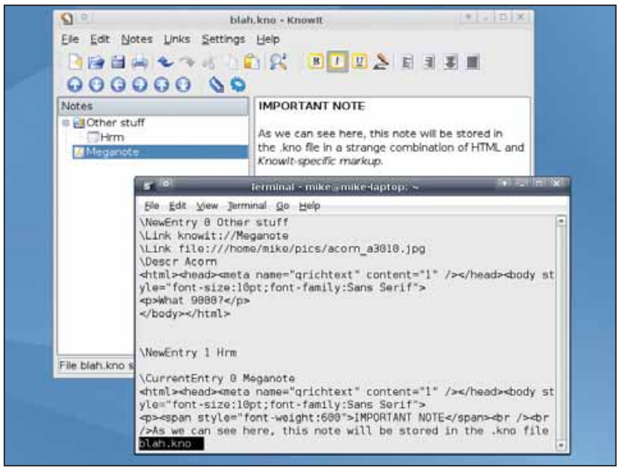
Изобилие буквы «K» в названиях программ KDE уже навязло в зубах, но в случае с *KnowIt* в ней, по крайней мере, есть смысл. *KnowIt* – менеджер заметок для KDE 3, который в последнее время особенно не разрабатывался – самая свежая версия, 0.10, вышла в марте 2004 года. Однако программа отлично работает и поэтому заслуживает нашего внимания в сегодняшнем обзоре. При первом запуске *KnowIt* появляется знакомое пользователям KDE окошко, где показаны различные настройки программы и комбинации клавиш.

Интерфейс битком набит кнопками, с которыми можно поиграть, но интуиция не поможет вам создать новую заметку: если кликнуть на кнопку New на панели инструментов, ничего не произойдет. Вместо этого нужно выбрать пункт меню Notes > Add и ввести название заметки.

Так что по части интерфейса дебют у *KnowIt* не блестящий, но стоит освоить основы, и все пойдет, как по маслу. *KnowIt* поддерживает обычное форматирование текста (полужирный шрифт, курсив, цвет, выравнивание) и позволяет добавлять ссылки на записи –текстовые поля, отсылающие к другим заметкам или файлам на компьютере. Самое заметное преимущество программы – функции сортировки: с помощью панели слева можно переупорядочивать заметки и даже создавать подзаметки. Это позволяет очень просто генерировать главные заметки для различных тем, а затем – дочерние заметки с конкретными данными.

Предусмотрен базовый экспорт в HTML, и, что любопытно, *KnowIt* сохраняет файлы в достаточно интересном сочетании «самодельной» разметки и HTML (см. рис.). Однако пока нельзя импортировать заметки из других программ, как нельзя и шифровать данные. А вот и плюс: в *KnowIt* можно перетаскивать текст из других приложений KDE без обычной бюрократии с копированием и вставкой. Это неплохая программа, но сейчас немного отстает в развитии.

> «В KnowIt можно легко перетащить текст из других приложений KDE.»



› С помощью панели слева можно перемещать заметки и создавать подзаметки в виде дерева.

LINUX FORMAT Вердикт

KnowIt

Версия: 0.10
Сайт: http://knowit.sourceforge.net
Цена: Бесплатно по лицензии GPL

» Программа устарела, и в ней маловато функций, но зато она симпатична и эффективна.

Рейтинг 7/10

NoteCase

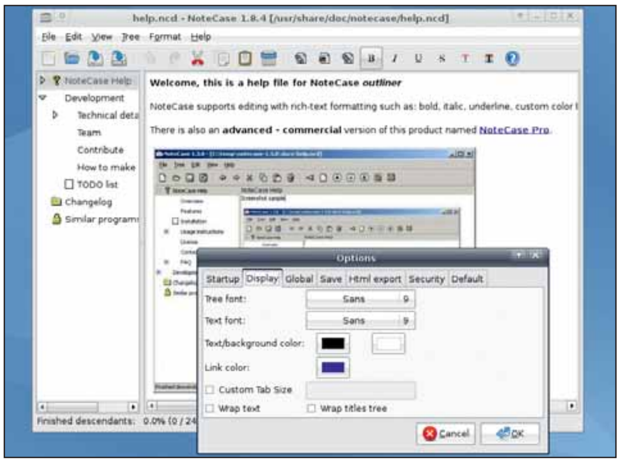
Выжать максимум из небольших мониторов.

Если зайти на сайт *NoteCase*, можно увидеть множество экранных снимков программы с мониторов с низким разрешением, на таких платформах, как Sharp Zaurus и Maemo. Интерфейс и правда очень компактный: меню и единственная панель инструментов. Как и в *KnowIt*, в *NoteCase* используется окно из двух панелей – на левой находятся названия заметок. Можно кликнуть по ним мышью, переупорядочить их и создать подзаметки в дереве. В *NoteCase* заметки называются «узлами» ('node'), но это не приводит к путанице.

В плане функций с *NoteCase* все хорошо: есть форматирование текста, встраивание картинок, счетчик слов и режим «только для чтения» (в нем невозможно случайно удалить данные). Опций очень много: можно задать сочетания клавиш и выполнить тонкую настройку запуска приложения и его внешнего вида, резервного копирования и экспорта в HTML. *NoteCase* умеет читать и сохранять файлы во многих форматах, в том числе .mm от Gjots, *Sticky Notes* и *FreeMind*; предусмотрено задание пароля и генерация зашифрованных файлов.

Самая полезная функция *NoteCase*s – флаг «завершенности» узла. Когда нужда в заметке отпадет, ее можно пометить как «завершенную», и сделать то же самое для прочих ненужных заметок – а потом одним ударом «зачистить» мусор. *NoteCase* умеет даже копировать собственный исполняемый файл, присоединив к нему текущий документ, и отправлять заметки вместе с программой вашему другу или коллеге, и им не придется устанавливать приложение самим. В целом, *NoteCase* – прекрасная программа со всеми необходимыми функциями, а сверх того и собственными изюминками.

> «NoteCase — прекрасная программа, имеющая все необходимое и более того.»



› Возможности программы демонстрирует документация, поставляемая в сложном файле *NoteCase*.

LINUX FORMAT Вердикт

NoteCase

Версия: 1.8.4
Сайт: http://notecase.sourceforge.net
Цена: Бесплатно по лицензии BSD

» Лучшая из предложенных программ на GTK: зрелая, полная возможностей и очень хваткая.

Рейтинг 8/10

Basket

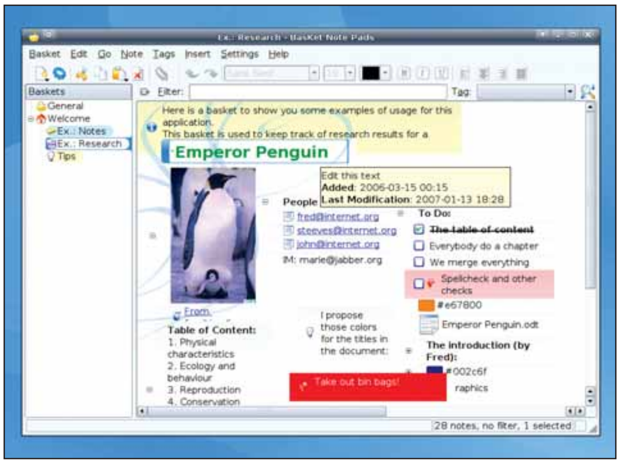
Скорее не «корзина», а фургон возможностей…

Ведь правда, KDE лучше Gnome?.. Шутка, шутка! Мы не будем разводить старый флейм (поэтому и взяли для нашего теста рабочий стол *Xfce*), однако, больше всего масла в огонь дебатов «KDE против Gnome» подливает старый показатель «число функций»: приложения Gnome в целом стремятся к простоте и минималистичному интерфейсу и не стараются впихнуть в себя все сущее, как мы видели на примере *Tomboy*. А вот для сторонников KDE огромное количество опций выбора и виджетов в порядке вещей. Для примера, сравните *Konqueror* и *Epiphany*.

Basket – приложение KDE, и, следовательно, это пиршество функций, с количеством кнопок, способным истрепать мышь за пару часов работы. Но, как отмечалось в предисловии, функции сами по себе – это не всегда хорошо, особенно если программа выглядит сложнее теории суперструн. На фоне остальных рассмотренных приложений, интерфейс *Basket*, безусловно, самый насыщенный. Есть меню, панель инструментов, панель навигации по заметкам, фильтр, список тэгов и собственно содержимое заметки. Также вы увидите приятную анимацию: фрагменты заметок грациозно соскальзывают на свое место.

В терминах *Basket*, корзина – это набор заметок, обычно с одним основным документом и несколькими дочерними; очень похоже на *KnowIt* и *NoteCase*, но в *Basket* эта концепция расширена: кликнув в произвольном месте основной панели заметки (справа), можно добавлять текст в маленькие блоки. Подготовив фрагменты текста, можно выделить и перетащить их, чтобы сгруппировать. Это дает больший контроль над иерархией заметок по сравнению с остальными программами; очень эффективная возможность, хотя и требует некоторого времени на освоение.

Впрочем, текстом дело не ограничивается: можно вставлять картинки, ссылки на web-страницы, флажки (для todo-списков) и ссылки для запуска программ. Слегка потрудившись, *Basket* легко превратить в стартовую площадку всех ваших дел: заметки, напоминания, ссылки и приложения хранятся в одном месте. Каждый фрагмент данных в *Basket*, будь то текст или картинка, допускает изменение размера, и его можно переместить; это очень похоже на издательскую систему. Благодаря продуманной функциональности, эта программа намного опережает соперников по обзору – она очень разносторонняя.



› *Basket* дает полную свободу в планировании своих заметок в стиле продвинутых издательских систем типа *Scribus*.

Эй, отлично выглядишь!

Анимация при перемещении заметок – чисто косметический эффект, и он также наблюдается во время поиска. После ввода текста в фильтр в верхней части окна фрагменты заметок с неуместной информацией исчезают из поля зрения. Да, это роскошь, но она подчеркивает внимание, уделенное разработке *Basket*. Любому фрагменту информации можно назначить предварительно созданные тэги (например, низкий приоритет, работа, личное) или создать собственные тэги со своими изображениями и цветами.

Экспорт в HTML весьма впечатляет. Создается web-страница (с изображениями), практически идентичная исходному документу (на уровне пикселей). Можно импортировать файлы из большого количества альтернативных программ, включая *KNotes*, *KJots*, *KnowIt*, *TuxCards* и *Tomboy*, можно защитить заметки паролем (а программа заодно оценит его стойкость). Интеграция с рабочим столом на должном уровне – *Basket* добавляет иконку в системный лоток, и картинки из *Konqueror* можно перетаскивать прямо в свои заметки.

> «Basket — достаточно гибкая, чтобы угодить и новичкам, и ветеранам.»

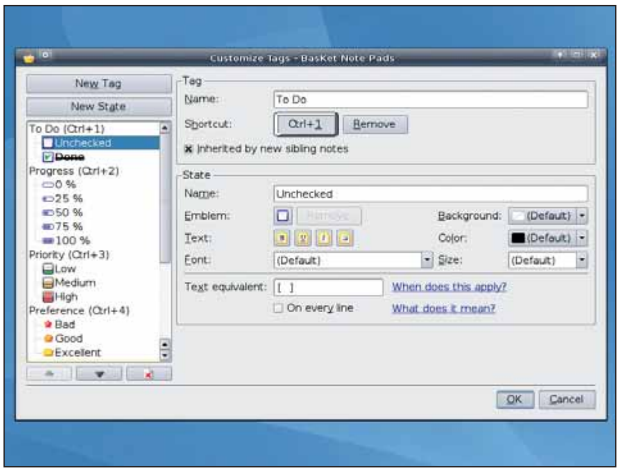
Перебор возможностей?

Итак, основной вопрос в случае с *Basket* таков: не перебор ли здесь функций? Ну, в 95% случаев – нет. Интерфейс программы вполне сравним с большинством участников теста, и ничего лишнего вам не навязывается. При всех возможностях добавления анимации, задания категорий, тэгов, цветов, шрифтов, картинок и прочих красот, можно обойтись и простым текстом, если вам его достаточно. Это признак хорошей программы – она достаточно гибкая, чтобы понравиться как новичкам, так и ветеранам. Некоторые спецы по интерфейсам говорят, что слишком много возможностей настройки интерфейса «под себя» – это плохо и нарушает целостность; другие возражают, что программа от этого более гибкая.

Конечно, пользователям Gnome, которым не нужны все эти причиндалы, идеально подойдет *NoteCase*, а в противном случае *Basket* – лучший выбор, самый гибкий и продуманный менеджер заметок, имеющийся в Linux.

Тэгтастика

В *Basket* есть масса заранее определенных тэгов, так что можно пометить конкретную заметку, скажем, как завершенную на 25% или просто забавную. Но тэги – еще и своего рода шаблоны: они способны влиять на форматирование фрагмента заметки. Например, можно установить тэг «срочно» ('urgent'), и он увеличит весь связанный с ним текст и выделит его красным цветом и жирным шрифтом. Тэгам можно даже назначить клавиатурные сокращения, чтобы быстро помечать фрагменты текста нужными тэгами. Большинству пользователей достаточно иметь пяток тэгов, но огромная гибкость реализации тэгов в *Basket* повышает его оценку в этом Сравнении.



› Одна из выдающихся черт *Basket* – это тэги. Вот диалоговое окно для создания тэгов и управления ими.

LINUX FORMAT Вердикт

Basket

Версия: 1.0.2
Сайт: http://basket.kde.org
Цена: Бесплатно по лицензии GPL

» *Basket полным-полна умными функциями; здесь есть все, чего можно желать от менеджера заметок.*

Рейтинг 9/10

TuxCards

Несомненно, это блокнот для фанатов Tux.

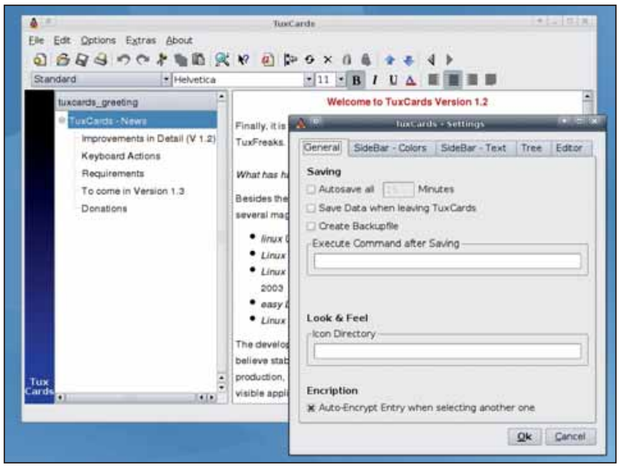
Все, что связано с нашим любимым пингвином-талисманом, сразу получает поощрение в виде поднятого большого пальца. Однако другой большой палец решительно смотрит вниз, пока мы не увидим, что же *TuxCards* может нам предложить. Из преимуществ: *TuxCards* поставляется в виде статически скомпонованного двоичного файла (куда включены все необходимые библиотеки), и он запустится практически на любом дистрибутиве. Из недостатков: по сравнению с другими приложениями он выглядит весьма бедно.

Как и во многих программах из этого обзора, в *TuxCards* двухпанельное окно. На левой панели – названия заметок, на правой – их содержимое. Слева можно упорядочить заметки в древовидную структуру: создать подзаметки и переместить их. Доступно базовое форматирование (стиль и размер шрифта, выравнивание и т.д.), но в текст заметки нельзя вставить картинки, и нельзя перетащить файлы из других приложений, чтобы появились внутренние картинки или ссылки. Если перетащить файл из *Konqueror*, в тексте заметки появится лишь **file://<имя_файла>**.

Импортировать данные тоже нельзя, но есть экспорт в HTML, а данные *TuxCards* хранит в четко читаемом XML-формате. Предусмотрено шифрование по паролю; при этом шифруется не весь файл, а только текст заметки в секции CDATA. Это довольно удобно – можно зашифровать лишь важные части заметок, без превращения всего файла в нечитаемый, если потом понадобится отредактировать его в текстовом редакторе.

TuxCards очень примитивен в сравнении с *NoteCase* или *Basket*, и кроме шифрования отдельных заметок, нет причин предпочесть его чему-то еще (хотя он на диво быстр). *TuxCards* – очень старая программа, которая почти не разрабатывалась с 2004 года; это не катастрофа, но она отстает в развитии от своих аналогов.

> «У TuxCards окно из двух панелей, как и у многих таких программ.»



› ***TuxCards* еще туда-сюда в плане настроек, но функций может предложить очень мало.**

LINUX FORMAT Вердикт

TuxCards

Версия: 1.2
Сайт: www.tuxcards.de
Цена: Бесплатно по лицензии GPL

» *При всей примитивности, TuxCards – неплохая программа, но она нуждается в обновлении, чтобы догнать конкурентов.*

Рейтинг 4/10

Zim

Это wiki… но на рабочем столе!

Мы следили за *Zim*, начиная с версии на Perl-GTK из HotPicks в LXF81. Тогда это была новинка, задумавшая ввести редактирование в стиле wiki в настольные приложения – в частности, была предложена идея создания страниц заметок на лету: вы набираете какой-то текст, выделяете его фрагмент и превращаете его в ссылку на новую страницу. И мы его зауважали, обнаружив, что подход wiki к управлению информацией прекрасно работает в случае с заметками.

Однако сейчас в большинстве тестируемых менеджеров заметок эти возможности есть, поэтому *Zim* не выглядит редкостью. Однако это неплохая легкая программа: текст можно редактировать сразу, доступно базовое форматирование, для создания новой страницы можно выделить слово или два с помощью мыши и кликнуть на кнопке ссылок на панели инструментов. Лучшая черта *Zim* – его формат хранения данных. Это не самопальный диалект XML и не нечитаемый двоичный формат: *Zim* использует wiki-подобный синтаксис. Так, например, выделенный жирным шрифтом текст хранится в виде '**жирный шрифт**'. Это позволяет довольно легко перемещать заметки между разными программами или просто редактировать их в текстовом редакторе в случае необходимости. Но кроме этого *Zim* предложить почти нечего.

Можно экспортировать заметки в HTML, но импортировать их из других программ нельзя, как нельзя и зашифровать и/или защитить их паролем. Тем не менее, для простых заметок этой программы более чем достаточно, и можно перетаскивать в свои заметки файлы и картинки, чтобы создать ссылки или внутренние изображения.

Если вам нужны по большей части текстовые заметки, связанные друг с другом без всякой системы категорий или иерархии, *Zim* – идеальный, легкий и приятный менеджер заметок; но не ждите от него большего.

> «Zim – идеальный, легкий и приятный инструмент, но большего не ждите.»



› **Обширная онлайн-документация *Zim* поставляется в формате Zim для демонстрации возможностей программы.**

LINUX FORMAT Вердикт

Zim

Версия: 0.23
Сайт: http://pardus-larus.student.utwente.nl/~pardus/projects/zim

» *Скромный по функциям и нетребовательный к памяти, Zim не может тягаться со «большими мальчиками», но основные потребности покрывает неплохо.*

Рейтинг 7/10

Менеджеры заметок

Вердикт

Basket 9/10

С самого начала казалось, что реальная борьба предстоит только между *Tomboy* и *Basket*. Первый – одна из самых известных программ Mono (поэтому мы и отдали ему целую страницу), и хотя мы наблюдали за ним не один год, но были удивлены тем, как слабо он подготовлен к конкуренции с KDE. Пока нас не завалили гневными письмами, спешим сказать: мы понимаем, что версия *Tomboy* – всего лишь 0.10.0. Мы не придираемся к нему из-за нехватки возможностей, но не любим, когда что-то вроде бы реализовано, но работает шатко, если вообще работает. Нет сомнений, что *Tomboy* 1.0 будет отличной программой, но сейчас ему есть куда развиваться.

Итак, *Tomboy* не взял первый же барьер, давая шанс *NoteCase* догнать *Basket* и посостязаться с ним. *NoteCase* – сказочная программа: море возможностей, прекрасно вписывается в Gnome и *Xfce* и не выказывает признаков нестабильности (даже когда мы вставляли в заметки множество случайных данных).

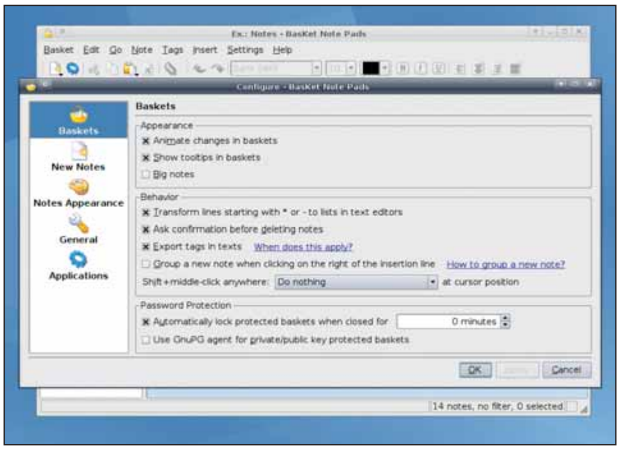
> «Чтобы удовлетворить все потребности, сразу установите Basket – и не разочаруетесь.»

Победитель только один

Но *NoteCase* – не конкурент *Basket*, когда дело доходит до полного набора возможностей. Способность *Basket* экспортировать заметки с попиксельной точностью (как будто мы работаем с настоящими листами бумаги) в сочетании с системой тэгов делает его почти непобедимой программой для управления заметками. Ее интерфейс кое-где требует подчистки, не допуская оценки 10/10, но это наверняка можно будет исправить в версии для KDE 4.

С учетом сказанного, итог таков: кому нужен лучший менеджер заметок с миллионом возможностей, удовлетворяющих любые потребности, просто установите *Basket* – и не разочаруетесь. Как KDE-приложение, он слегка в диссонансе с Gnome или *Xfce*, но стоит этой жертвы: вы влюбитесь в его гибкость. Но если возможности *Basket* кажутся излишними и вы регулярно работаете в Gnome или *Xfce*, попробуйте *NoteCase*. Хотя его возможности – капля в море по сравнению с *Basket*, *NoteCase* очень разносторонний, и у него приятный интерфейс.

В целом это хорошая презентация для сообщества Open Source: при любом рабочем столе, сколько возможностей вы ни хотели бы иметь под рукой, всегда найдется достойное приложение. Будем ждать версии *Basket* для KDE 4 и некоторых улучшений в *Tomboy*.



› Масса функций, приятная наружность и высокая настраиваемость *Basket* завоевали высшую оценку в этом тесте.

Обратная связь

Вам попадался менеджер заметок и получше? Считаете, что *NoteCase* для психов, а *Basket* обвешан KDE'шными рюшечками? Никогда не расстанетесь с ручкой и бумагой? Для нас всегда ценно ваше мнение – напишите нам на letters@linuxformat.ru или присоединяйтесь к разговору на наших форумах www.linuxforum.ru.

Таблица свойств

Название	Версия	Размер исполняемого файла	Время запуска, сек[1]	Использование памяти, МБ	Иконка в системном лотке	Импорт[2]	Внутренние изображения	Шифрование
Basket	1.0.2	45k	7	18MB	✓	7	✓	✓
KnowIt	0.10	238k	5	16MB	☒	0	☒	☒
NoteCase	1.8.4	1.1MB	2	15MB	✓ (optional)	4	✓	✓
Tomboy	0.10.0	318k	10	35MB	✓	0	☒	☒
TuxCards	1.2	723k	4	13MB	☒	0	☒	✓
Zim	0.23	7.9k	3	27MB	✓ (plugin)	0	✓	☒

[1]«Холодный» (программа не кэширована в ОЗУ) запуск в Xfce с незагруженными библиотеками Gnome или KDE, так что никаких предубеждений
[2]Сколько форматов файлов можно импортировать, не считая собственного формата данных

Все программы тестировались на компьютере с процессором 1,6 ГГц Celeron M с 1 ГБ ОЗУ, под Xubuntu 7.10

Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux



ЛАДИСЛАВ БОДНАР
основатель, редактор, начальник и сотрудник DistroWatch.com.

Стабильность против новизны

Какой дистрибутив вы выберете в середине 2008 года? Реальных прорывов нет ни у одного, и выбор будет во многом определяться вашими личными пристрастиями. Предпочитаете ли вы хорошо протестированную стабильную ОС с минимумом поправок после выпуска? Или вы в восторге от суперпродвинутых технологий, даже если они иногда приводят к сбоям в работе?

Среди четырех главных настольных дистрибутивов имеется явное разделение на эти две группы. Mandriva и Ubuntu приняли более консервативный подход при выборе пакетов для версий 2008.1 и 8.04, и хотя оба предоставляют KDE 4.0 в качестве дополнительной опции, они более озабочены стабильностью пакетов и спокойной работой, чем предоставлением наисвежайших, но потенциально проблемных программ.

Новые разработки

С Fedora и OpenSUSE все не так. Оба эти дистрибутива сделали стандартным рабочим столом KDE 4.0, а OpenSUSE к тому же переработал свой установщик и инструмент настройки под обновленный инструментарий Qt. Конечно, рабочим столом по умолчанию в Fedora остается Gnome, но при текущей тенденции делать KDE равным игроком в дистрибутиве, полное отсутствие KDE 3.5 в Fedora 9 удивляет. Одно из преимуществ OpenSUSE над Fedora – увеличенный период разработки, поскольку релиз OpenSUSE 11.0 назначен не ранее середины июня, через полтора месяца после Fedora 9.

Если ваш приоритет – стабильность и отлаженный набор приложений, выбирайте Mandriva 2008.1 или Ubuntu 8.04. Но если вам подавай новейшие технологии, то Fedora 9 и особенно OpenSUSE 11.0 – лучший выбор.

ladislav.bodnar@futurenet.co.uk

Hardy Heron

Ubuntu 8.04 LTS – виртуализация, Gnome 2.22, новые стандартные приложения, установщик Wubi и так далее.

С выпуском в конце апреля версии 8.04, Ubuntu продолжила свой уверенный марш к настольному Linux и доминированию в серверном сегменте. Уже во второй раз к названию продукта был добавлен значок LTS [долгосрочная поддержка]. Второй LTS-релиз Ubuntu означает, что Hardy Heron [Стойкая Цапля], кодовое имя версии 8.04, будет поддерживаться на уровне исправлений безопасности и ошибок на рабочих столах – до апреля 2011, а на серверах – до апреля 2013. Стал ли новый Ubuntu хорошей системой для развертывания на офисных компьютерах и корпоративных серверах?

Хотя сведения о релизе выделяют некоторые новые функции, пристальный взгляд обнаружит, что Canonical переключила свое внимание на улучшение серверной версии продукта. В конце концов, именно на этом компания наиболее вероятно сможет заработать, продавая поддержку в стиле бизнес-модели Red Hat. Отлаженный продукт и обещание долгосрочного обслуживания могут привлечь куда больше клиентов, чем предыдущая LTS-версия 6.06 два года назад.



› **Ubuntu Hardy Heron положил глаз на бизнес-рынок, но 8.04 пленит новыми функциями и домашних пользователей.**

Основные нововведения в Ubuntu 8.04 Server – виртуализация, интеграция ActiveDirectory, поддержка iSCSI и механизмы защиты памяти. Первая обеспечивается KVM [Kernel-based Virtual Machine, Виртуальная машина ядра] и включает графический инструмент управления *Virtual Machine Manager*, разработанный Red Hat. Это одна из наиболее востребованных функций для многих high-end провайдеров, и Ubuntu 8.04 LTS теперь имеет хорошие шансы для полноценной конкуренции с Red Hat и Novell на этой арене.

> «Canonical переключила внимание на улучшение серверной редакции.»

Развернувшие новый Ubuntu на настольном компьютере также не будут разочарованы. Хотя 8.04 – преимущественно стабилизирующий релиз, пользователи найдут достаточно причин для обновления. Среди них – последняя версия Gnome (2.22), свежее настольное ПО, интересные приложения и новое оформление. Но самая замечательная функция – это *Firefox 3.0* с функцией Places, новой инфраструктурой для унифицированного хранения закладок, истории и другой информации о web-страницах.

В Ubuntu 8.04 также интегрирован *PulseAudio*, набирающий популярность звуковой сервер, плюс *PolicyKit*, инструмент для тонкого контроля за выполнением программ и доступа даже к приложениям, обычно требующим привилегий суперпользователя. Есть еще *Vinagre*, новый VNC-сервер, позволяющий работать одновременно с несколькими удаленными компьютерами. Другой новинкой является инсталлятор *Wubi*. Доступный на LiveCD в качестве исполняемого файла Windows, он предоставляет простой способ установки и удаления Ubuntu как очередного приложения Windows – без привлечения каких-либо ресурсов жесткого диска и смены загрузчика.

www.ubuntu.com

Офис в сети

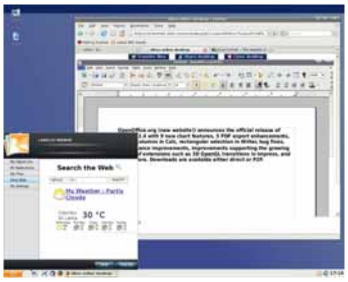
Ulteo Application System – создаем, редактируем и публикуем документы без проблем.

Когда Гаэль Дюваль [Gaël Duval] приступил к разработке нынешнего Mandriva Linux в далеком 1998, у него была идея, оказавшаяся выигрышной. Он взял самый популярный дистрибутив Linux того времени и упростил его инсталляцию и настройку, добавив графический установщик, удобные конфигураторы и более интуитивный рабочий стол.

Десять лет спустя основателя Mandrake осенила новая интересная идея: на сей раз – сделать наши файлы и данные универсально доступными из любой точки мира и независимо от используемой на компьютере ОС.

Первая бета-версия Ulteo была выпущена в конце марта. Хотя она доступна в виде установочного LiveCD, Ulteo не стоит рассматривать как очередной дистрибутив, конкурирующий с Mandriva или OpenSUSE – и уж точно он не апеллирует к фанатам ОС, жаждущим новейших технологий.

Вместо этого Ulteo сосредоточен на сетевых сервисах и групповой работе. По сути, чтобы стать пользователем Ulteo, нет нужды даже устанавливать его на компьютер, поскольку многие службы проекта уже доступны пользователям в любом web-браузере с поддержкой Flash и Java.



› Любой зарегистрированный пользователь сервисов Ulteo может создавать и редактировать документы *OpenOffice.org* в сети.

Тем не менее, Ulteo (дистрибутив) упростит создание, редактирование, совместное использование и извлечение ваших документов, сохраняемых на удаленном сервере, в частности, посредством вкладки My Digital Life в стартовой панели Ulteo и его функции автоматической синхронизации. Окончательная версия также обещает предоставление метода безопасного шифрования для документов, хранящихся в сети.

Судя по его энергичному росту, Ulteo имеет потенциал стать весьма притягательным продуктом.

www.ulteo.com

Скромный запрос

Slitaz GNU/Linux 1.0 – самый маленький дистрибутив Linux в мире.

В мире, где системные требования растут с каждым новым выпуском Windows или MacOS, приятно встретить проект, не настаивающий на покупке дополнительной памяти или более мощной видеокарты.

Наоборот: Slitaz GNU/ Linux, новый дистрибутив, достигший версии 1.0 в конце марта, работает потрясающе быстро даже на машинах далеко не первой молодости. При наличии 128 МБ ОЗУ и привода CD-ROM, вы сможете наслаждаться полнофункциональным рабочим столом с браузером, мультимедийными и графическими приложениями, и web-сервером с интегрированной поддержкой CGI и PHP.

Slitaz GNU/Linux – LiveCD, который загружается в память, запуская легковесный оконный менеджер JWM. В меню – два браузера (*Firefox* и *Retawq*), аудиопроигрыватель (*AlsaPlayer*), CD-риппер (*Asunder*), графический редактор (*MTPaint*), клиенты FTP (*GFTP*) и BitTorrent (*Transmission*), почтовый клиент и просмотрщик PDF.

Если всего этого мало, то гораздо больше доступно для загрузки посредством Tazpkg, схожего с APT менеджера пакетов.



› Сколько приложений можно вместить в 25 МБ? Оказывается, вагон...

Как это принято у современных LiveCD, Slitaz можно установить на жесткий диск или USB-носитель несколькими щелчками мыши. Консольная утилита для ремастеринга LiveCD также прилагается. А главное, Slitaz 1.0 поставляется как ISO-образ в 25 МБ, так что даже пользователи с телефонным соединением смогут загрузить его довольно быстро.

Но истинно впечатляет в Slitaz GNU/Linux то, что разработчики смогли уместить так много функционала в столь малом объеме. Действительно, жемчужина среди дистрибутивов!

www.slitaz.org

Шпаргалка по продуктам Ubuntu 8.04

Из одиночного установочного CD с универсальным дистрибутивом Ubuntu разросся в линейку продуктов, обладающих большим разнообразием сценариев развертывания. Некоторые из них – издания Ubuntu с официальной поддержкой, другие сопровождаются небольшими группами сторонних разработчиков, но всегда есть одна общая черта: ядро Linux и Ubuntu как основа системы. Для тех читателей, кто ищет себе самый подходящий продукт, приводим краткий обзор с их описанием и формой распространения.

	Цель издания	Доступен в виде
Kubuntu-KDE4	Ubuntu с новейшей версией KDE 4.0	Установочный CD, LiveCD
Edubuntu	Ubuntu для школ; содержит образовательное ПО	CD с дополнениями
Gobuntu	Вариант Ubuntu, включающий только свободное ПО, по определению FSF	Установочный CD
Ubuntu Server	Издание Ubuntu для серверов, поддерживается пять лет	Установочный CD
Ubuntu JeOS	Минималистичный Ubuntu, разработанный специально для виртуального применения	Установочный CD (только i386)
Ubuntu Studio	Содержит большую коллекцию аудио-, видео- и графических приложений	Установочный DVD
Mythbuntu	Минималистичный Ubuntu с MythTV и соответствующим ПО	Установочный CD

Хит-парад дистрибутивов

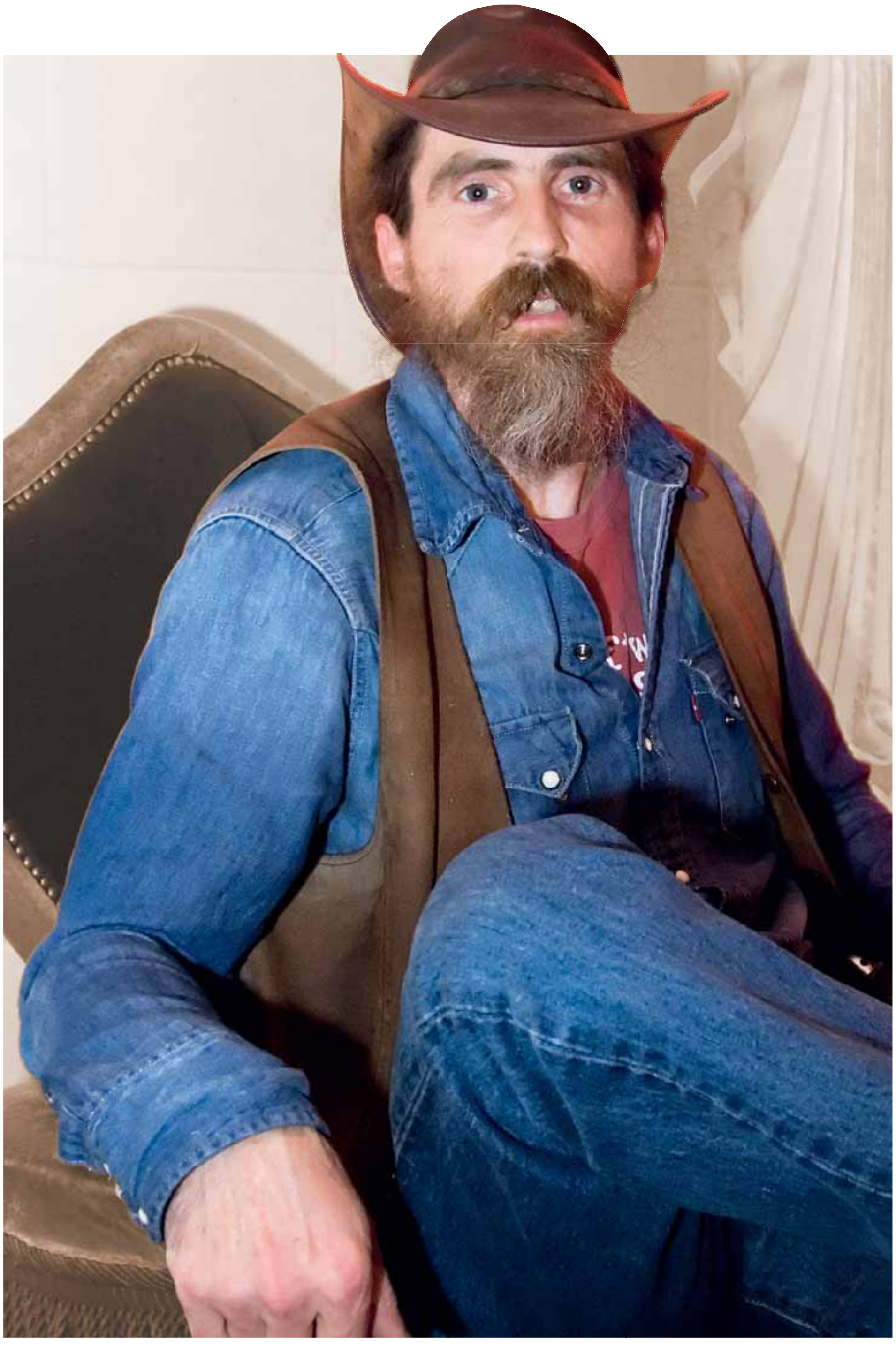
10 самых посещаемых страниц на DistroWatch.com с 4 апреля по 2 мая 2008 (среднее число визитов в день)

	Дистрибутив	Число визитов	
1	**Ubuntu**	3,702	↔
2	**OpenSUSE**	1,765	↑
3	**Fedora**	1,578	↔
4	**Mint**	1,351	↓
5	**Mandriva**	1,349	↑
6	**PCLinuxOS**	1,092	↓
7	**Debian**	841	↑
8	**Kubuntu**	796	↑
9	**Slackware**	743	↑
10	**Dreamlinux**	716	↑

» **DistroWatch.com** оценивает популярность дистрибутивов, основываясь на числе визитов на страницу каждого из них. Хотя это не дает представление о действительном числе установок, но показывает, какие дистрибутивы более популярны за определенный промежуток времени. LXF

Быстрый, как Gnash

Adobe разрабатывал Flash десять лет, а вот Робу Савою понадобилась лишь 1/10 этого времени. **Майк Сондерс** беседует с маэстро кодирования…



Отнюдь не каждый способен на обратный инжиниринг самого популярного в Интернете проприетарного формата. Опять же, отнюдь не каждый умеет программировать с GCC, GDB, OLPC, Cygwin, DejaGnu и Linux, находя при этом время для помощи в ликвидации последствий урагана в Новом Орлеане. А вот Роб Савой [Rob Savoye] делает все это — и даже больше, и его свободная версия Adobe Flash Player настолько обогатила состав открытых кодеков, что у web появился шанс на подлинно свободное будущее.

Linux Format: **В порядке вступления, не могли бы вы вкратце поведать нашим читателям историю вашей работы в мире свободного ПО?**

Роб Савой: Над свободным ПО я начал работать в 1987 году, в проекте GCC… Тому уже 20 лет! Я работал над GCC и GDB по заказу одной компании, производившей оборудование: ей были не по карману дорогостоящие компиляторы от Sun Microsystems. Так что свободное ПО помогло нам в те дни. Итак, пару лет я работал над GCC и GDB, и к концу 80-х нам наприсылали столько заявок на платную поддержку, что в 1989 мы основали Cygnus Support. Я работал в Cygnus Support почти до момента ее продажи Red Hat примерно в 2001. Так что, можно сказать, я занимаюсь только свободным ПО уже 20 лет. Я немало потрудился над GCC и GDB, я написал *DejaGnu*, написал *libgloss*, работал над *newlib*, начинал проект *Cygwin*, и я разработал операционную систему eCos. С 1991 года плотно работаю с Linux, и уже много лет консультирую НАСА и других в области GCC и ядра Linux. А потом я начал работу над проектом Gnash в качестве пользовательского интерфейса для ТВ-приставок.

LXF: **Погодите, так вы перешли с GDB на Gnash? Вот это прыжок!**

РС: Мой друг Джон Гилмор [John Gilmore] вложил средства во много мелких компаний. И вот однажды он звонит мне и говорит: «Эй, я знаю людей, которым нужен интерфейс пользователя на основе Flash для их стерео», а я ему: «Отлично, я за цифровую музыку, с ней веселее!». Поискал я в Интернете, что там есть насчет свободных программных плейеров – оказалось, не так уж и много. Ну, я взял оттуда технологию, добавил всякий там ActionScript – и сделал интерфейс для стерео. Проект был сроком на полгода, и я и думать о нем забыл – он же был разовым, верно? А месяца через два снова звонит мне Джон: «Эй, а как ты смотришь на переделку того Flash-плейера в подключаемый модуль для Firefox?» Отвечаю: «Джон, я с графикой не работаю, и потом, я пакую вещи ехать в Новый Орлеан, на ликвидацию последствий Катрины [сильнейший ураган, фактически уничтоживший этот город, – прим. пер.]». А вернувшись с Катрины через пять месяцев, я созрел – позвонил Джону и сказал: «Да ведь Flash-плейер в Firefox – это здорово!» И я его сделал. И на нас мгновенно обрушилась извест-



ность, просто потому, что он заработал. Тогда у него не было поддержки YouTube, но для многих из нас, сторонников свободного ПО, он был важен, и благодаря моему опыту работы в команде GCC Gnash работает на 64-битных машинах, на других архитектурах – это много, много больше, чем умеет версия от Adobe, и я вдруг понял, что существует насущная потребность в свободной Flash-технологии, и рамки этого проекта расширились настолько, что охватили и работу над непатентованными кодеками. Но так уж вышло – Gnash подобен множеству проектов, которые вначале не планировались, просто так вышло. А потом я понял, что хорошие проекты сами вас находят.

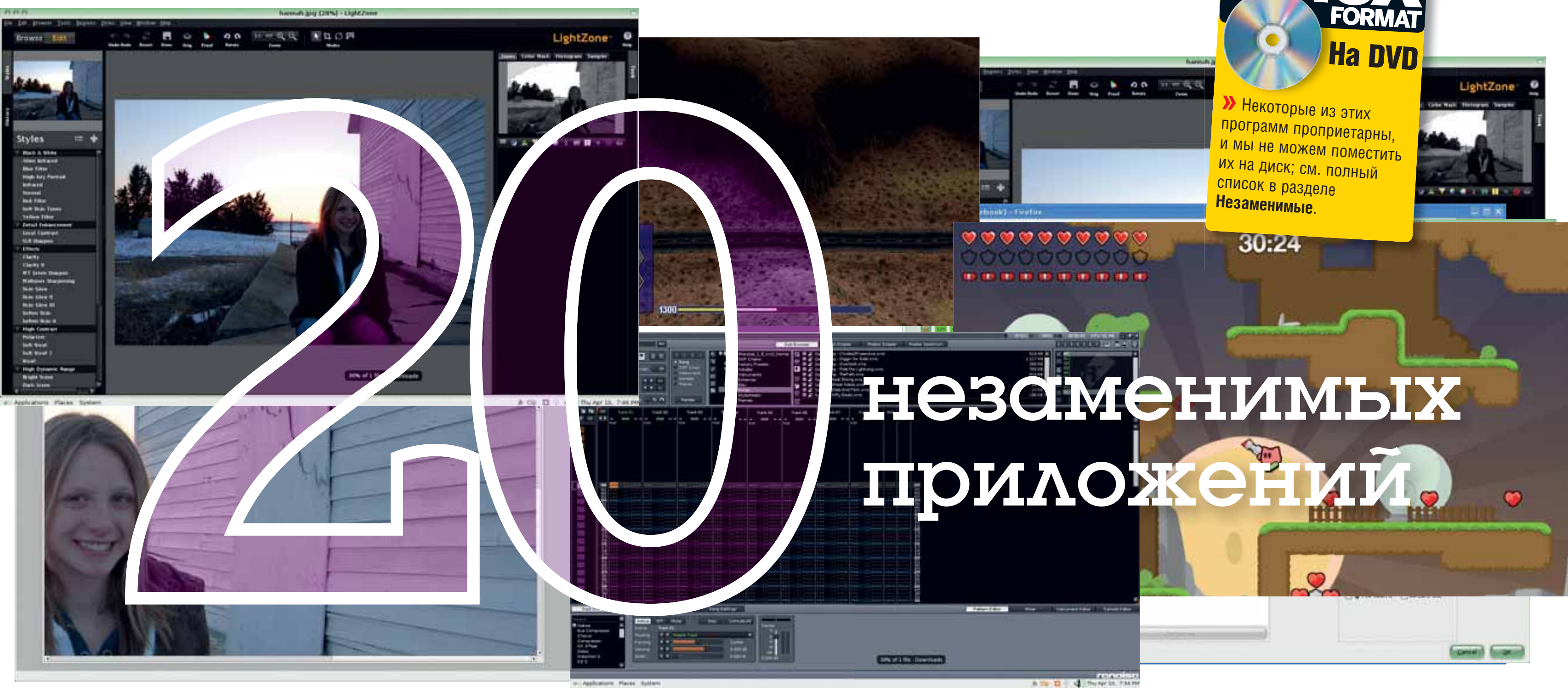
LXF: Вы по-прежнему занимаетесь GCC или GDB?

PC: Слегка. Я много работал с GCC по проекту OLPC. Выполнил всю работу по оптимизации компилятора для чипа Geode GX. Хакеров-фрилансеров, работающих с GCC, не так уж много! Но вообще-то я интересуюсь встраиваемыми системами, так что я из тех, кто помогал добавлять в GCC поддержку кросс-компиляции; когда-то я даже делал первый Linux-порт на 68K. А потом я занялся Gnash – финансировали эту работу в основном я сам и Джон Гилмор, а то и вовсе никто не финансировал, а потом так вышло, что мне позвонил Боб Янг [Bob Young] (бывший глава Red Hat) и сказал: «У меня сейчас небольшой бизнес с прессой, и мы хотим по-настоящему взяться за цифровые мультимедиа. Нам нужно свободное ПО, а я знаю, что если вам подкинуть денег, вы сможете платить своим разработчикам, сможете работать полный рабочий день и сможете ускорить темп разработки, а моя компания сможет всем этим воспользоваться». И мы согласились. И вот года полтора Боб нас спонсировал. Но недавно мы осознали, что этого маловато, так как охват нашей работы расширился: сюда включилось то, что мы называем открытыми мультимедиа; Gnash – это только их часть. И мы создали некоммерческий фонд под названием Open Media Now (openmedianow.org), Боб и Джон вошли в совет директоров, и мы общаемся со многими компаниями, никогда раньше не включавшими в свои системы поддержки Flash. Если вы используете оборудование на основе MIPS или SH, экстравагантные чипы ARM, 64-битные PowerPC, то это все мы. У нас даже есть поддержка PS3! Мы в Itanium, мы и на Solaris, мы работаем на многих интересных ОС. Мы реально заботимся о многоплатформенности, а вот Adobe об этом вообще не думает.

> О НАЧАЛЕ ПРОЕКТА
> «Gnash сперва в плане не было, просто так вышло. А потом я понял: хорошие проекты сами тебя находят».

LXF: А почему вы решили, что Gnash уже недостаточно, и надо продвигать открытые мультимедиа?

PC: Да все из-за такого пустячка под названием «патенты на ПО». Если вы используете MP3 или FLV, они проприетарные. И хотя мы используем FFMPEG и Gstreamer – мы поддерживаем все эти кодеки – мы не можем таким образом распространять Gnash. Так что меня завалили отчетами об ошибках – о том, что YouTube не работает; и меня это бесит. Даже когда мы работали над программой OLPC «обмен баш на баш», я тоже тонул в сообщениях об ошибках типа «А я-то думал, что на этой штуке YouTube работает». Но, естественно, проект OLPC не мог на законном основании распространять кодеки. Итак, меня забросали отчетами об ошибках, и мы вдруг поняли, что проблема, которую мы пытались решить, заключалась вовсе не в обратном инжиниринге Flash-плейеров, а в мультимедиа как таковых. Мы много работаем над тем, чтобы сделать их открытыми – Gnash полностью поддерживает такие непатентованные кодеки, как Ogg Vorbis, и Theora, и Dirac, и прочие. Но проблема была куда шире, и мы осознали необходимость создания целостной инфраструктуры: свободные кодеки существуют уже не первый год, но вечно получается так, что, скажем, проект поддерживает кодек, а сервер не поддерживает… его там просто нет. А если мы намерены сделать мультимедиа свободными, нужна целостная инфраструктура. Коммерческим компаниям дела до этого нет – они все лицензируют, но большинство наших друзей относится к миру свободного ПО, и мы хотим поддержать тех, кто выбрал Fedora, Ubuntu, Debian, BSD, и всех остальных. Wikipedia скоро откроет сайт для обмена видео, полностью на базе Ogg Vorbis и Theora, и для него нужен сервер. Сейчас мы работаем над видеоконференциями на Flash, а это – серверная сторона, все форматы тут проприетарные. Но если у вас есть сервер и есть клиент, можете спокойно сменить кодек, это никого не касается. LXF



LINUX FORMAT На DVD

» Некоторые из этих программ проприетарны, и мы не можем поместить их на диск; см. полный список в разделе **Незаменимые**.

20 незаменимых приложений

Если вы новичок в Linux, вас удивит изобилие свободных программ, включенных в ваш дистрибутив по умолчанию – и вы совсем уж изумитесь, поняв, что на свете их много больше. **Джон Брендон** сформировал горячую двадцатку незаменимых приложений...

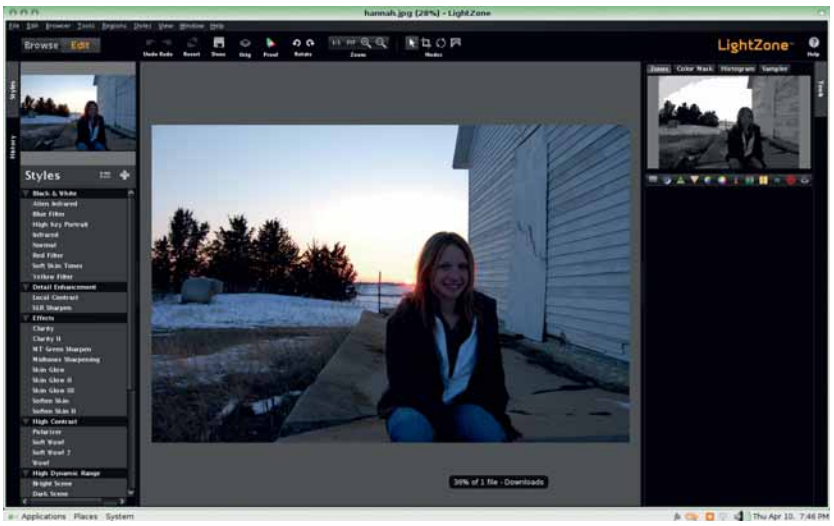
Графика

LightZone 3.4

www.lightcrafts.com/linux

Мощный инструмент редактирования фотографий, помогающий разобраться с подсветкой практически любого снимка Его особенность – «умный» интерфейс, обычно – с одним ползунком для выдержки, источников света с широким динамическим диапазоном [HDR], сепии и сотен других настроек, призванных улучшить изображение – или сделать его более креативным.

LightZone творит чудеса с темными снимками: он исправляет оттенки и повышает яркость каждого отдельного пикселя, и при этом фотография не выцветает. Бета-версию можно попробовать свободно, но окончательный релиз обойдется в $199,95.



› *LightZone* работает в согласии с теорией света и тени Ансела Адамса [Ansell Adams].

Qtfpsgui

qtpfsgui.sourceforge.net



Давние пользователи Linux знают, что основное преимущество этой системы – в том, что программист может решить сделать какой-то инструмент – как правило, для включения в собственный арсенал – и выпустить его на свободу. *Qtfpsgui* – именно такой инструмент. Он позволяет экспериментировать с HDR-освещением на фотографии или 3D-изображении, что можно использовать в играх и 3D-мирах. Органы управления просты: перемещаем ползунок для коррекции гаммы, загружаем инструмент настройки карт оттенков для коррекции источников света и других переменных. Затем сохраняем нашу работу для моделирования в 3D-мирах.

Hugin

http://hugin.sourceforge.net

Эксперименты с фотографией могут привести к чудесным результатам. *Hugin* – редактор фотопанорам, умеющий «сшивать» ваши снимки. Процесс таков: при съемке вы размещаете объектив для каждого очередного снимка так, чтобы правый край текущего снимка совпадал с левым следующего – при каждом новом снимке смещаясь вправо. Вам не требуется абсолютная точность, поскольку *Hugin* хорошо сшивает фотографии вместе – впрочем, штатив отнюдь не помешает.

Мультимедиа

Renoise

Не пропустите!

www.renoise.com

Выходя за пределы простой механики записи звука, создания аудиоклипа и выпуска его в свет, эта мощная студия создания музыки предназначена для желающих выполнить тонкое редактирование частот каждой новой композиции, не впадая в подход «удачной палитры» инструментов вроде *Apple Logic Pro*. Работая в Linux, она поддерживает библиотеки VST-инструментов, которые начали свое существование в Windows.

Встроенный сэмплер позволяет создавать уникальные звуки, а затем добавлять их на временную шкалу. Редактор шаблонов (паттернов) поможет создать из этих звуков повторяющуюся последовательность. Ползунки для создания (микширования) музыки (перенос влево или вправо, добавление эквалайзера и изменение громкости) помогут вам завершить свое произведение.



Alexandria

http://linux.softpedia.com/get/Desktop-Environment/Tools/Alexandria-3488.shtml

Если ваша офисная библиотека перегружена книгами, попробуйте *Alexandria*, утилиту работы с каталогами: она облегчает просмотр корешков, поиск книг, которые вы отдали на руки, или тех, которые следует выбросить, потому что они лишь занимают место. Хорошим источником обложек является Amazon.com, но убедитесь, что вы щелкнули на изображении, чтобы получить его полноразмерную версию. Затем просто сохраните его в каталоге и загрузите изображение в *Alexandria*. Можно осуществлять быстрый поиск по заголовкам и авторам, но основная цель *Alexandria* – напомнить вам, какие книги у вас есть и что еще бы вы хотели бы прочесть.

Jokosher



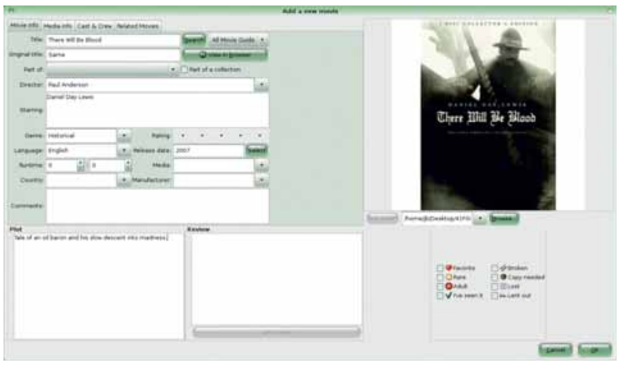
www.jokosher.org

Большинство линуксоидов знают об *Ardour*, мощной (но сложноватой) многотрековой программе для звукозаписи. (Пока что мы не готовы назвать ее станцией по работе с цифровым звуком.) *Jokosher* – более отлаженная программа записи треков для тех, кто решил создать свои демо-ролики. У нее мало функций для создания циклов или использования файлов, созданных с подключенной к ПК фортепианной клавиатуры, но она позволяет вам добавлять звуковые файлы прямо в тот же интерфейс, в котором вы записываете живые инструменты.

Каталог фильмов CeeMedia

ceemedia.sosdg.org

Подобно *Alexandria*, утилите одной задачи, программа *CeeMedia* является – как следует из названия – инструментом каталогизирования фильмов, которые у вас есть или которые вы уже посмотрели (или хотите посмотреть). Это занятная программа: она позволяет добавлять всякие данные о каждом фильме, включая актеров и состав команды, краткое содержание и даже мини-обзор. Мы считаем, что следующим хорошим шагом в разработке *CeeMedia* будет создание некой социальной сети, чтобы пользователи могли обмениваться рейтингами фильмов и обзорами – подобно тому, что сейчас делает Flixster.com. Понятно, что *CeeMedia* – это вообще-то оболочка базы данных для всех ваших видеозаписей, и в этом она превосходна.

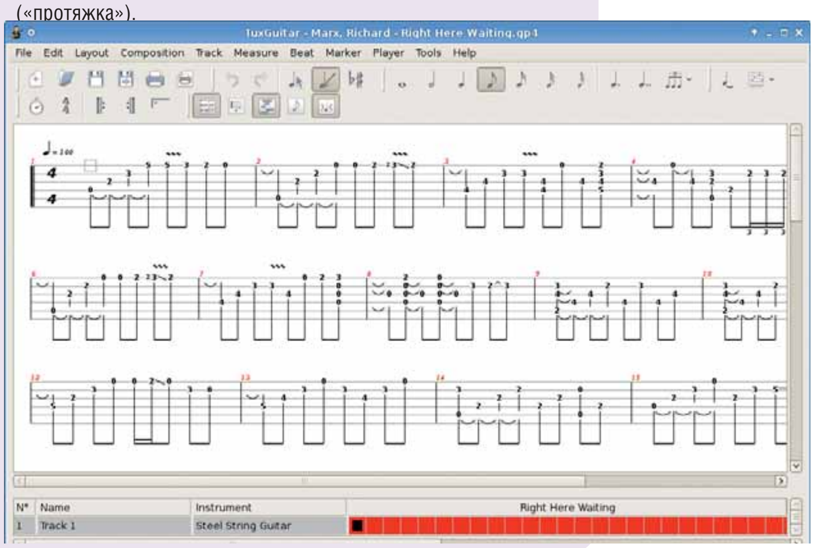


TuxGuitar

www.tuxguitar.com.ar

Запись партитур для гитар – нот и инструкций, помогающих другим гитаристам выучить песню – может быть трудной задачей. Обычно это требует использования особого шрифта в *OpenOffice.org* и рисования таблатур вручную, или использования дорогой программы создания музыки, предоставляющей сотни функций помимо простого создания гитарных партитур.

TuxGuitar предназначена только для создания таблиц: она лишена любых функций линейного упорядочения и записи и обеспечивает возможность создания партитур для любой песни. Вы можете отрегулировать темп, длительность нот, знаки, триоли и добавить к музыке такие эффекты, как вибрато и бенд («протяжка»).

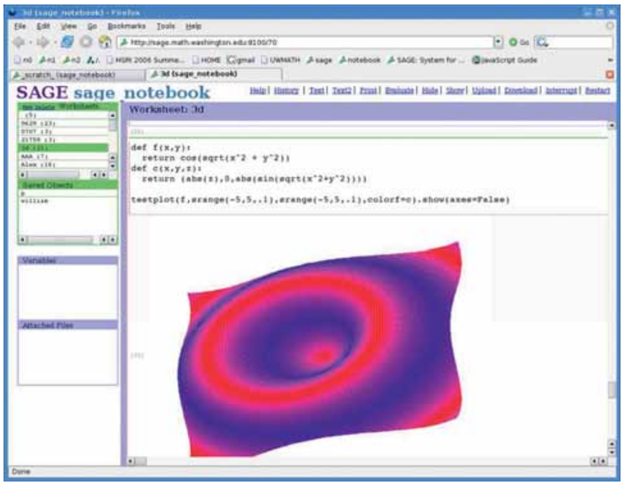


Офис

Sage

www.sagemath.org

Mathematica от Wolfram Research – все еще явный лидер программ «научных вычислений» на рынке конечных пользователей и образования. Приложение стоит несколько тысяч долларов, имеется Linux-версия, но вы можете делать практически то же самое с *Sage*, эквивалентом с открытым кодом. Нам показалось, что установить *Sage* сложновато: он даже не заработал на одном из наших ноутбуков с Ubuntu и имеет несколько неуказанных зависимостей, таких как *LaTeX*. Но если *Sage* запустится, вы получите набор функций для операций высшей математики, криптографии и анализа. На www.sagenb.org имеется онлайн-демонстрация, позволяющая опробовать приложение до установки.



› *Sage*: математическое приложение с открытым кодом для тех случаев, когда *OOo Calc* уже не справляется.

Dia

www.gnome.org/projects/dia

Как и более известный инструмент рисования *Visio* (работающий только в Windows), *Dia* разработана для простых блок-схем, поясняющих элементы сложных задач и планов. Она чуть менее специализированная, чем *Visio*, поскольку содержит несколько другой набор пиктограмм и изображений. Имеются все основные инструменты рисования – стрелки для формирования древа решения, несколько фигур для добавления в диаграмму и текст с поддержкой многих шрифтов и форматирования. Программа предстает в полном блеске, когда вы используете ее для UML-диаграмм, показывающих развитие процесса разработки приложения, а добавление цвета в эти утилитарные рисунки оживляет их. [*Dia* содержит множество библиотек иконок, не только UML, а вот отсутствие функции выгрузки кода делает ее практически бесполезной в разработке, – *прим. пер.*] *Dia* хорошо поддерживается сообществом, но пора бы и переходить к версии 1.

VariCAD 2008

www.varicad.com

Сложные CAD-программы, как правило, не дешевле рук, ног и селезенки, но *VariCAD* порывает с обычной ценовой схемой, упирающейся в тысячи. За $620 программа повторяет функции, имеющиеся в *AutoCAD*, поддерживает многие форматы файлов, такие как STEP и DWG, и огромные библиотеки существующих объектов (в основном разработанные для проектирования механизмов). Она также обладает интерфейсом, стремящимся разместить каждую иконку и опцию прямо в главном окне, чтобы они оказались на расстоянии одного щелчка. Это полезно для серьезных дизайнеров, потому что означает отсутствие охоты за шестигранным карандашом.

Развлечения

TeeWorlds

www.teeworlds.com

Кто сказал, что стрелялка должна быть кровавой и жестокой? *TeeWorlds* – союз *Quake* и *Joust* (заключенный на небесах), где вы бегаете по платформам и отстреливаете других сетевых игроков. В ней есть выключатели и различные наземные и подземные карты, объекты, заграждающие ваш путь, а также мультяшная графика, и она позволяет другим игрокам сшибать вас с дистанции. Эта игра, которую следовало бы назвать *TeeWars* – теперь с открытым кодом, хотя сперва была коммерческим проектом. Она чрезвычайно затягивает; уникальный проект, совмещающий два уникальных стиля игры.



The Sims Carnival

www.simscarnival.com

Мы включили сетевой игровой портал, чтобы показать, что имеются сотни, если не тысячи, игр, доступных сообществу Linux в браузере, независимо от наличия родной Linux-версии. Одним из новейших является The Sims Carnival, предлагающий прорву аркадных игр, головоломок и даже игр-приключений. Самое интересное в Carnival то, что другие пользователи создают игры для вас, так что это превосходное введение в основы программирования: игровой процесс, графика, дизайн и тестирование.

Cuyo

www.karimmi.de/cuyo

Этот отпрыск *Tetris* работает потому, что он очень прост и совершенно лишен 3D-блоков. Вместо них *Cuyo* представляет мультяшные улыбающиеся лица, скатывающиеся с травяного холма. На поздних стадиях игра ужасно усложняется, поскольку вы пытаетесь сочетать цвета и формировать цепочки лиц, но лица падают с разной скоростью, а некоторые заставляют соседей взрываться. Другая уловка в том, что иногда вы должны сформировать строку улыбающихся лиц по диагонали до того, как они взорвутся, а иногда спускающиеся лица меняют цвет, и на это следует быстро реагировать.

Stellarium

Не пропустите!

www.stellarium.org

Известно, что Google создал программу отображения карты звездного неба *Google Sky*, а *Stellarium* – ее эквивалент с открытым кодом, привлекающий хорошей поддержкой сообщества и интерфейсом, делающим упор на разглядывание звезд, а не на традиционные (и сложные) функции планетария вроде азимутальной сетки и скриптинга координат. На самом деле программа повторяет большинство функций *Google Sky*, включая звездную сетку – и позволяя увидеть 600 000 звезд – но прежде всего она просто показывает широкий обзор неба для поощрения творчества и экспериментирования.



Warzone 2100

www.wz2100.net

Warzone 2100 основана на стратегии реального времени, разработанной Pumpkin Studios в далеком 1999-м. В 2004-м издатель-гигант Eidos Interactive опубликовал исходный код для сообщества, и родилась *Warzone 2100*. Игровой процесс следует традиционной механике *Command & Conque*, с напряженным сбором ресурсов и тактикой яростного огня, используя графический движок, который явно выдает свой возраст. Зато *Warzone 2100* свободна и имеет более глубокую древовидную структуру – с 400 технологиями для исследования и разветвленной структурой настройки, тремя кампаниями и 24 картами мгновенных действий.

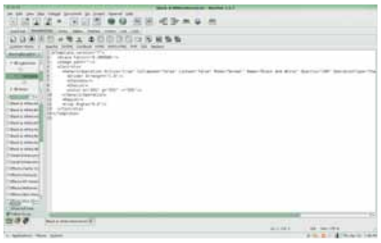
› Похоже, на данном экранном снимке *Warzone 2100* плохиши все разбежались.

Разработка

Bluefish

bluefish.openoffice.nl

Новый с иголочки продукт в версии 1.0, *Bluefish* – мощный текстовый редактор для программистов. Заявленный как «чрезвычайно легкий», он использует лишь половину памяти, требуемой большинству хорошо известных редакторов вроде *Quanta*, и прекрасно сможет работать в легковесном дистрибутиве – например, Fluxubuntu – или вместе со средами программирования типа *Eclipse*. К тому же он быстро загружает файлы: в тесте мы открывали десять HTML-файлов практически в мгновение ока, хотя они были достаточно сложны.



Salasaga

www.salasaga.org

Хотя Adobe недавно разродилась инициативой Labs и поддержкой Linux для все большей части своих компьютерных продуктов (таких как *Flex* и платформа Air), все еще практически невозможно найти версии ее основных приложений, таких как *Dreamweaver* или *Fireworks*, запускаемых в Linux. *Salasaga* заполняет этот дефицит, позволяя создавать анимированые Flash-ролики в родном формате SWF, и в конечном итоге будет поддерживать Ajax для полностью интерактивного содержимого в браузере.

Octave 3

www.gnu.org/software/octave

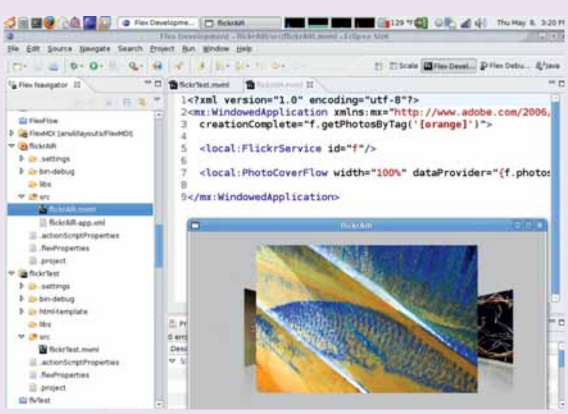
Эта платформа разработки призвана помочь программистам с вычислительными задачами – включая нелинейные уравнения и полиномы, работу с дифференциальными-алгебраическими уравнениями и другими сложными математическими задачами – посредством собственного интерфейса программирования. Основное улучшение версии 3.0 – это лучшее взаимодействие с *Matlab* и инструмент анализа и визуализации данных, помогающие программистам со сложными технологиями для высокотехнологичных разработок.

Adobe Flex Builder Public Alpha

Не пропустите!

www.adobe.com/products/flex

Напоследок! Эта среда разработки для *Eclipse* – она работает как модуль расширения – позволяет создавать web-приложения даже без подключения к сети, как *Google Gears*. Основной упор сделан на легкой разработке скриптов для максимально быстрого создания качественных приложений Web 2.0 с использованием отлично переносимого языка MXML, интуитивных инструментов разработки интерфейса и открытых стандартов. LXF



Обратная связь

Нашли ли вы приложение, без которого не можете жить? Опробовали ли вы все из перечисленных выше и поняли, что им нет места в вашей лодке? В любом случае мы хотим знать об этом. Пришлите ваше незаменимое приложение на letters@linuxforum.ru или опубликуйте ваши находки на форуме LXF по адресу www.linuxformat.ru.

Райский рабочий стол

Хотите сделать его человечнее? Надоел синий цвет? **Грэм Моррисон**, наш штатный эстет, опишет форму и возможности рабочего стола Linux.



Мы проводим много времени за рабочим столом Linux. Это наш офис, гостиная, игровая комната — все в одном. Как и в этих помещениях, пребывание здесь становится гораздо приятнее, если обстановка комфортна и подстроена под привычный вам способ работы. Это виртуальный эквивалент перестановки дивана, чтоб он был перед теликом, помещения пива в холодильник и зажигания пары свечей.

Может, это прозвучит странно, но наш рабочий стол – это выражение нас самих и того, как мы используем компьютер. Именно поэтому Интернет забит людьми, которые выкладывают изображение своего рабочего стола; доходит даже до разработки собственных наборов значков и написания приложений, позволяющих в корне все изменить. Некоторые изменения – только для красоты, например, фоновый рисунок или цветовая схема. Но есть и другие, отражающие порядок вашей работы: значки приложений в панели инструментов, сочетания клавиш для их запуска, и даже размер шрифта и расположение окон.

> «Рассмотрим те области, где малые изменения ведут к большим результатам.»

На последующих страницах мы рассмотрим основные области рабочего стола, где малые изменения приводят к большим результатам, и охватим главные графические окружения, используемые большинством из нас. Авось это вдохновит вас на самостоятельные перемены!

Под микроскопом:

Xfce
Им часто пренебрегают, однако это окружение рабочего стола стоит взять на заметку.

KDE
Забудьте синий цвет, меняйте краски и выкиньте из головы все мысли о том, что это «круто»

Gnome
Превратите Gnome в эквивалент мягкой подушки с помощью опций, о существовании которых вы и не подозревали.

Compiz
Глазурь на пирожном. Compiz может украсить любой рабочий стол, и не только напоказ.

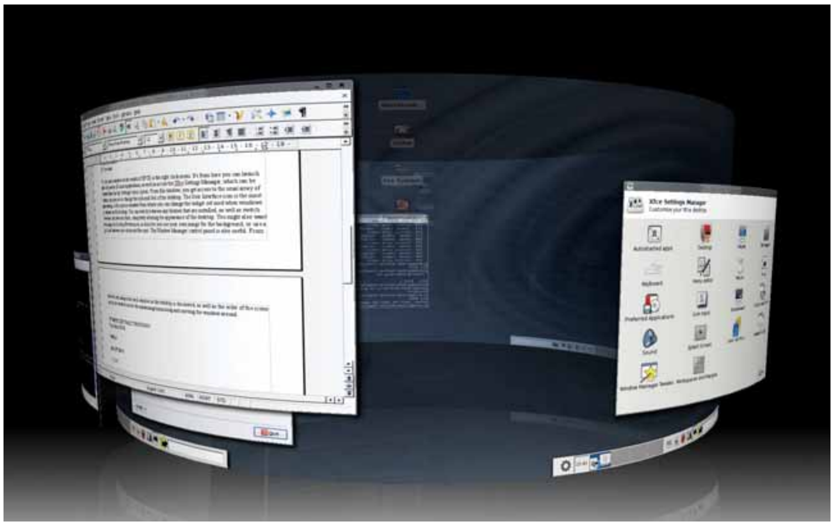


› **Нудный старый KDE... Не волнуйтесь, мы его улучшим.**

Xfce

Начнем с фундамента: с менеджера рабочего стола, который не говорит вам, как нужно использовать ваш собственный компьютер.

Дистрибутивы часто игнорируют *Xfce*, козыряя последними версиями KDE и Gnome, но для модификации рабочего стола *Xfce* – лучшее начало: ведь по умолчанию он предоставляет лишь пустое поле. Это может отпугнуть людей, поскольку на первый взгляд *Xfce* чересчур спартанский и нелюдимый. Но, заглянув «под капот», вы поймете, что *Xfce* – один из самых настраиваемых из существующих рабочих столов Linux. При желании его легко подогнать под Gnome или KDE, надо только потратить на это немного времени.



› *Compiz* используют, чтобы придать окружению рабочего стола современный вид, и *Xfce* – не исключение.

Ветер перемен

Начать следует с маленького значка в форме звезды, расположенного на рабочем столе. Это *Xfce* Panel, и она такая же, как панели других рабочих столов. Единственная разница – по умолчанию на ней нет никаких значков. Поправим это: щелкнем правой кнопкой по звездочке и выберем Add New Item. Откроется окно, отображающее все доступные апплеты. Например, вы можете добавить Часы [Clock], чтобы узнавать время. Другим хорошим выбором будет апплет Icon Box – он показывает значки для всех свернутых приложений. Desktop Pager поможет узнать, на каком из рабочих столов вы находитесь; вы всегда можете переключаться между ними с помощью комбинации Ctrl+Fn. Скоро вы сумеете в точности воссоздать панель, используемую вами на другом рабочем столе. Вы даже можете загрузить Gnome'овские апплеты с помощью *XfApplet*; есть и другие, более творческие возможности – например, встроенная командая строка *Verve* или апплет *Smart Bookmark*. Можно создать ссылки на ваши любимые приложения с помощью значка Launcher, можно изменить внешний вид панели, щелкнув по ней правой кнопкой и выбрав Customise Panel. Открывшийся диалог позволяет задать размер панели и указать, к какой стороне экрана она будет прилегать. Чтобы передвинуть панель, протащите ее мышью за край.

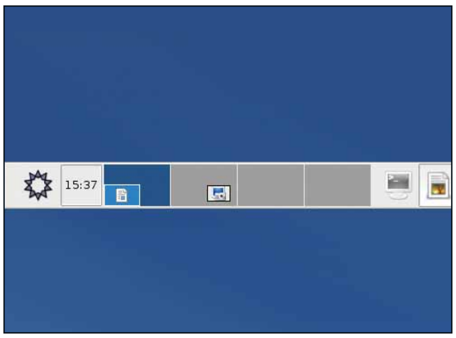
Главная дверь в мир *Xfce* – контекстное меню. Именно отсюда запускается обычный набор Linux-приложений, а также предоставляется доступ к *Xfce Settings Manager*, который запускается через верхний пункт меню Settings. Из этого окна вы можете получить доступ к обычному набору настроек, используемых для изменения внешнего вида рабочего стола.

В меню

Значок User Interface – самый интересный: он открывает окно, где можно поменять набор виджетов, используемый при прорисовке окон на экране. Вы можете переключаться между темами, набором значков и шрифтов, полностью преображая внешность *Xfce*. Возможно, вы также захотите зайти в Desktop Preferences – это позволит вам использовать отдельный фоновый рисунок для каждого рабочего стола, или же градиентный переход от одного цвета к другому. Также полезна панель управления Window Manager. Оттуда можно сменить оформление каждого окна на рабочем столе, а также порядок следования значков максимизации/минимизации и перемещения окна.

»

Три главных совета: Xfce



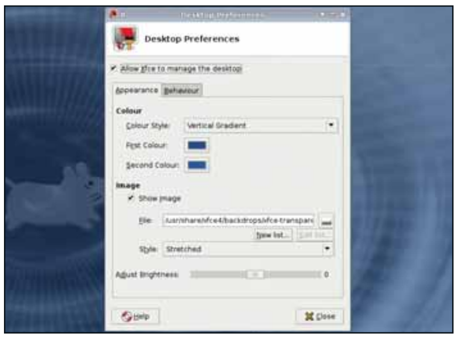
1 На панель

Начните с добавления на панель нескольких новых функций. Это поможет *Xfce* больше походить на рабочий стол, к которому вы привыкли.



2 Темы

Поменяйте голую тему, стоящую по умолчанию, на что-нибудь поинтереснее. Есть даже темы, выглядящие точь-в-точь как приложения *Qt 4*.



3 Фон

Для рабочего стола используйте градиентный фон или изображение вместо однородного цвета по умолчанию у многих дистрибутивов.

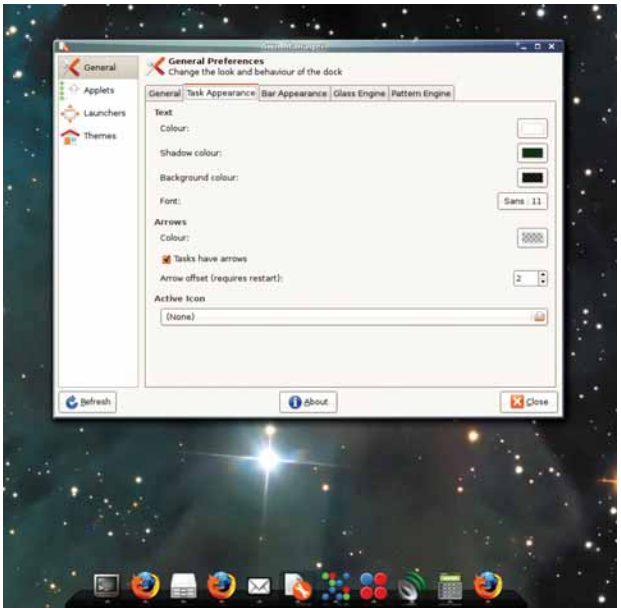
»

Отделка Gnome

Роднее Gnome места нет.

Справедливо будет утверждать, что Gnome, устанавливаемый по умолчанию в большинстве дистрибутивов, имеет более осторожный подход к потенциалу изменения рабочего стола, чем KDE. Он обычно предлагает отлаженный пользовательский интерфейс, с гораздо меньшим числом опций.

Вы также обнаружите две панели – менеджер задач внизу экрана и меню в самом верху; очень похоже на рабочий стол OS X. Если вам не по душе такой подход, одну из панелей можно убрать, щелкнув по ней правой кнопкой и выбрав Remove, но нам скорее нравится иметь обе панели. По ряду причин, верхняя панель, с часами и меню Places, кажется, сочетается с философией Gnome, и на современных дисплеях хватает места на обе.



› Gnome'овский *Avant Window Navigator* – наилучшая реализация дока в стиле OS X, когда-либо виденная нами в Linux, и наилучшая замена нижней панели Gnome.

Сногсшибательные значки

Слишком много тем по умолчанию используют коричневый и оранжевый (или как там они сейчас называются в Ubuntu) цвета. Вы можете поменять весь рабочий стол, просто переключив цветовую схему на использование синего и зеленого. Изменения такого типа производятся в Gnome'овской панели настройки Appearance (System > Preferences). Страница по умолчанию отображает установленные темы. Это – просто украшения рабочего стола, значки и цвета, упакованные в единые пакеты, которые можно быстро переключать. Любой из них легко отредактировать, нажав кнопку Customise: откроется окно, где следует указать свои сочетания для каждого элемента.

Но лучшее в окне Gnome Appearance – это поддержка drag and drop. Вы можете добавлять новые элементы и темы, скачивая пакеты и перетаскивая их в окно Appearance. Элемент будет автоматически установлен и активирован, и в списке появится новая Custom-тема. Это дубликат текущей темы с вставленным в нее новым элементом. Например, если вы скачиваете с www.gnome-look.org новый набор значков и перетаскиваете полученный архив в окно, то он добавится к списку доступных и будет активирован автоматически. Это очень упрощает пробы новых тем – вот лучший способ перестроить рабочий стол на свой лад.

> «AWN идеально заменит Gnome'овскую нижнюю панель задач.»



› Gnome-овское окно Appearance поддерживает перетаскивание, так что пользоваться им сможет и ребенок.

Панельные приложения

В Gnome есть парочка замечательных виджетов, которыми можно заменить нижнюю панель задач. Простейшим в использовании и установки является *SimDock*. Вызванный из меню Accessories, он прикрепится к нижнему краю экрана, сразу с масштабируемыми значками, представляющими запущенные на данный момент приложения. Но *SimDock* еще немного сыроват. Он копирует задний план и использует его в качестве фона для собственных рисунков, причем не всегда аккуратно. К счастью, есть решение получше. *Avant Window Navigator* (*AWN*) дает пользователям Linux наилучшее представление об Apple-овском Dock, а кое-где и превосходит его. Единственный его недостаток – *AWN* требует композитного менеджера, чтобы смешивать фон и отражения на экране в реальном времени. Это значит, что для его работы нужен запущенный *Compiz*, но игра стоит свеч – в особенности, если вы и так используете *Compiz*.

AWN также идеально заменяет Gnome'овскую панель задач (ту, что внизу экрана), так как воспроизводит ее функциональность и занимает меньше места. При первом старте он отображает все значки приложений, открытых на вашем рабочем столе. Запустив утилиту *awn-manager* или щелкнув по панели задач правой кнопкой, вы увидите окно настроек. В нем вы сможете добавлять ссылки на новые приложения, а также менять внешний вид панели. Нам кажется, что эффект 3D Turn выглядит великолепно, хотя у него есть склонность пожирать ресурсы CPU, когда вы тащите указатель мыши над панелью.

Чтобы серьезно расширить функциональность рабочего стола Gnome, следует подумать о добавлении апплетов. Это небольшие самостоятельные приложения, которые выполняют определенную задачу, не обременяя вашу систему полноценными приложением. Апплеты



› Это наша тема «Mushy Pea [Мутный горох]», с собственными значками и цветовой палитрой (дайте нам знать, если вы хотите, чтобы мы ее выложили).

бывают двух основных типов. Первые – панельные, и их можно добавлять только к панели Gnome (верхней или нижней). Апплеты второго типа обитают на рабочем столе и функционально очень похожи на Apple'овские виджеты Dashboard. В Gnome они называются *GDesklets* (декслеты).

Копаемся в виджетах

Перед тем, как начать, неплохо бы продраться сквозь ваш менеджер пакетов и установить все апплеты, что вам приглянулись, так как часто нельзя сказать, что делает данный апплет, пока его не запустишь. Чтобы добавить апплет, щелкните правой кнопкой по панели и выберите пункт Add to Panel. Как и в KDE, откроется окно, где отображены все апплеты, совместимые с Gnome. Наши предпочтения включают следующее:

» **System Monitor** Горизонтальная гистограмма использования CPU/памяти/диска/сети. Щелкните правой кнопкой, чтобы появилось окно настроек, где можно поменять цвет каждой компоненты.

» **Desktop Drapes** Скажите панели настройки Drapes, где вы храните обои рабочего стола, и они будут автоматически меняться каждые 15 минут (по умолчанию).

» **Drawer Секретное** хранилище для других апплетов, скрывающихся за значком Drawer. Хорошее место для запретных удовольствий типа апплетов Eyes или Fish.

» **Dictionary Look up** Если ваш словарный запас невелик, введите слово в текстовое поле ввода, и виджет отыщет его определение в свободной версии английского онлайн-словаря Вебстера.

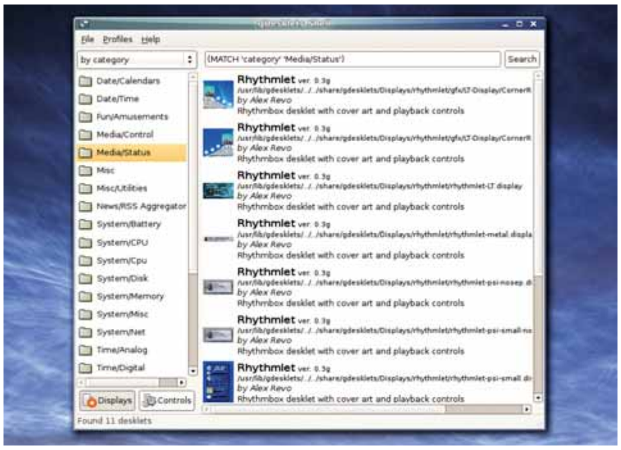
GDesklet'ы



Панельные апплеты хороши для небольших задач, вроде показа даты и времени, но они не годятся, когда вам нужно больше места. Решение состоит в том, что программисты окрестили *GDesklet*. Это апплет, который сидит на фоне рабочего стола, подобно *SuperKaramba* и Plasma в KDE, и виджетам Dashboard в OS X. По умолчанию они не являются частью рабочего стола Gnome; это значит, что их следует устанавливать и запускать отдельно из Gnome'овского меню Applications > Accessories. В области апплетов на панели задач возникнет маленький значок, и, щелкнув по ней правой кнопкой и выбирая Manage Desklets, можно добавить к рабочему столу новые виджеты. В появляющемся окне слева показан список категорий – выбор одной из них наполнит список десклетов справа.

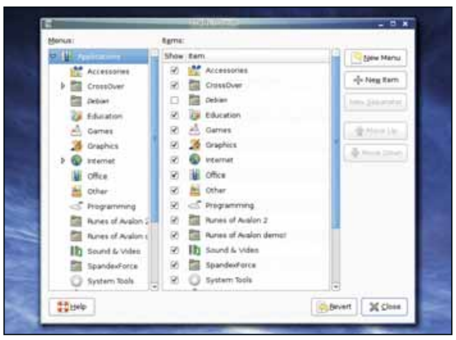
Многие апплеты, например, календари, менеджеры ресурсов, датчики температуры и разнообразные цитаты дня, включаются по умолчанию. Дважды кликните по тому, что вам нужно, и апплет прилепится к указателю мыши; останется лишь переместить его туда, куда требуется. Апплеты прикованы к своим позициям, но их можно сдвинуть, если щелкнуть правой кнопкой мыши или левой, удерживая правый Alt. Как и при почти любой модификации рабочего стола, лучшим подходом является метод проб и ошибок.

»



› GDesklet'ы похожи на KDE'шные апплеты *SuperKaramba*, и для их выбора есть тьма опций.

Три лучших совета: Gnome







1 Расчистка меню

Очень скоро меню будет переполнено приложениями. Отбракуйте ненужные с помощью System > Preferences > Main Menu.

2 Обои

Существует множество сайтов, способных дать вам вдохновение для обоев рабочего стола, но наши любимые – NASA и National Geographic.

3 Фон и эмблемы

Одна из редко используемых функцией файлового менеджера Gnome – способность метить определенные файлы эмблемами и менять фоновый рисунок при обзоре файлов.

KDE стал круче

Красота – это всего лишь форма, но что если вам не нравится синий цвет?

KDE можно считать самым прозаичным из всех общепринятых окружений рабочего стола. Гляньте на экран по окончании свежей установки: стандартная панель старомодно растянута по всей нижней границе экрана, а меню приложений уже десять лет обитает в левом нижнем углу. В зависимости от выбранного вами дистрибутива приложения, отображаемые в меню, располагаются как попало, что может затруднить поиск нужной вам программы.

Такой подход не привел в KDE новых друзей за последние пару лет, и даже KDE 4 не сильно изменил внешний вид. Проблема в том, что многие давние пользователи KDE долго и тщательно создавали себе удобное окружение, и они просто переносят свои настройки с одной установки на другую, редко когда пытаясь поставить чистый KDE, благодаря тому, что почти каждая опция для каждого приложения KDE сохраняется в подкаталоге **.kde** домашней папки. Он частично повторяет директорию, обычно расположенную в **/usr/share/kde**. Если опция существует локально, она перекрывает заданную глобально – и вот так вы копируете ваши параметры от одной установки на другую.

Внешний вид

Решившись изменить внешний вид вашего рабочего стола, откройте KDE'шный диалог System Settings. Это окно – просто рог изобилия опций настройки, но не будем отвлекаться – идем в раздел Look and Feel. Именно отсюда можно поменять такие вещи, как цветовая схема, шрифты и стили, используемые для отображения окон KDE, причем разъяснений не потребуется. Наш выбор лучших опций – переключение на стиль Klear looks, который эмулирует классический вид Gnome с похожим именем, и использование оформления окон Plastik – наименее неприятное из доступных вариантов. Смена цветовой схемы тоже может помочь, но в KDE темы значков часто склонны использовать синий цвет, и сложно отыскать другую схему, которая гармонировала бы с панелью или значками рабочего стола. Значки, конечно, тоже легко поменять, но, на наш взгляд, набор Crystal самый лучший, если только у вас не KDE 4: в нем используется Oxigen, потомок Crystal. Вы даже можете установить значки Oxygen в KDE 3. Просто скачайте пакет с www.kde-look.org, переключитесь на страницу Icons в окне Appearance в System Setting и выберите скачанный архив. KDE установит значки в локальный каталог KDE, и их можно будет активизировать из той же страницы Icons, хотя для того, чтобы увидеть эффект полностью, придется перезапустить KDE.



› **Изменение стиля виджетов в KDE влияет на способ отображения различных компонентов окон. Пробуя стили, пользуйтесь предпросмотром в Control Centre.**

Панели

Другой возможностью рабочего стола KDE, которой часто пренебрегают, является KDE Panel, чья гибкость позволяет приспособить ее почти к любой задаче. Самое частое добавление – апплеты панели. Это маленькие функциональные значки, которые тихо сидят себе в углу экрана. К таковым относятся, например, часы, хотя их настройки по умолчанию грубоваты. Вы легко можете сделать часы более эстетичными, щелкнув по ним правой кнопкой и выбрав Configure Clock. Попробуйте сменить тип на Plain Clock, задействовать рамку и поменять цвет фона на более темную градацию серого.

Чтобы добавить новые апплеты, щелкните по пустому месту панели и выберите Add Applet to Panel, но сперва вам лучше установить все то, что есть в менеджере пакетов вашего дистрибутива. Рекомендуем попробовать:

» **KNewsTicker** Читалка RSS-лент, прокручивающая через низ экрана заголовки статей. Для открытия соответствующей страницы щелкните по заголовку. Просто потрясающе, сколько можно прочесть по ходу другой работы.

» **System Monitor** Три крохотных вертикальных или горизонтальных линии, представляющие использование CPU, свободной памяти и файла подкачки. Существенны при поиске неустойчивых процессов и глючных версий Firefox.

» **KWireless Monitor** Показывает уровень сигнала и список локальных точек доступа вашего беспроводного интернет-соединения. Выберите какой-нибудь из них, чтобы подключиться к сети.

» **KNetload** Рисует диаграмму, иллюстрирующую количество входящих и исходящих данных вашего сетевого соединения. Полезно, если за трафик платите вы.

» **KSim** Системный монитор типа «все в одном». Он занимает немного больше места на экране, зато на панели можно наблюдать список процессов, температуру оборудования и другую статистику.

KDE 4

Если у вас нет желания марать руки, то для изменения внешнего вида KDE 4 сделать можно не особо много. Но если вы не боитесь редактирования файлов конфигурации, то сможете поменять изображения, используемые панелью.

В KDE 4 есть достаточно возможностей для подстройки окружения на ваш вкус. Первый кандидат – это виджеты Plasma, так как интерфейс их добавления встроен в рабочий стол. Но их слишком мало, чтобы можно было выбирать, и пока популярность KDE 4 не вырастет, ситуация вряд ли изменится. Все еще есть планы встроить в виджеты Plasma совместимость с Dashboard, что придаст рабочему столу KDE 4 огромный потенциал, но пока это не заработает, вам придется довольствоваться весьма скучным набором, включенным в поставку. В окне настройки KDE 4 меньше опций Look and Feel, чем то изобилие, что имеется в KDE 3, и панель рабочего стола не такая гибкая, как в третьей версии. С другой стороны, здесь все еще есть *Konqueror*, самый настраиваемый менеджер файлов, и набор значков Oxygen заставит сердце биться сильнее.





› **Меню KDE может быстро замусориться. Рекомендуем заменить меню по умолчанию на Tasty Menu: оно легче воспринимается и имеет собственный поиск приложений.**

Деление панелей

Панели можно помещать в любое место экрана, причем в произвольном количестве. Щелкните правой кнопкой по любому пустому месту, выберите Add New Panel – и вы увидите список из шести типов панелей. Dock Application Bar годится для приложений *WindowMaker* (конкурирующий менеджер окон). External Taskbar откроет новую панель выбора задач, где вы увидите все запущенные приложения и сможете переключаться между ними. Это удобно, если у вас есть второй экран: тогда вы можете позволить этой панели занять всю его нижнюю часть. Следующим пунктом после External Taskbar будет Kasbar. Первоначально он был разработан как замена панели задач, но сейчас это чуть более приятная панель, расположенная на экране вертикально справа. В традиционной панели трудно упорядочить значки и приложения по вертикали (даже если вы перетащите панель на другой край экрана), а вот Kasbar создает более удобные меню и значки. Но ничто не сравнится со стандартной панелью. Вы можете добавлять их сколько захотите, либо плавающие, либо прилепленные к краю экрана, и каждая новая панель будет ничуть не хуже самой первой. Это значит, что вы можете добавлять апплеты и панели задач, и рассматривать их как расширение исходной. Мы бы порекомендовали изменить размер панели либо на маленький, либо на крохотный, и даже поменять ее длину, чтобы она не простиралась на весь нижний край экрана.

Приложение KDE, наиболее близкое к Mac OS X Dock, называется *KoolDock*. Оно устанавливается посредством менеджера пакетов вашего дистрибутива, и когда вы запустите его в первый раз, понадобится указать несколько опций настройки. *KoolDock* отображается в нижней части экрана и по умолчанию выставляет только шесть значков. Значки меняют размер по мере того, как вы кружите над ними, а док сам по себе скрывается под нижней частью экрана через пять секунд. Его можно воскресить, подвинув указатель мыши к этому краю. В установках *KoolDock* можно включить менеджер задач, а также и меню приложений KDE. Альтернативно, рекомендуем получать доступ к меню Application по правому щелчку мыши на рабочем столе, чего можно достичь из страницы Desktop > Behaviour > General в KDE Control Centre.

> «Ближе всех к Mac OS X Dock приложение KDE под названием KoolDock».

SuperKaramba

Несомненно, *SuperKaramba* – король апплетов рабочего стола. Он породил целое поколение подражаний и стал исходным вдохновителем Plasma из KDE 4. Ничто не может соревноваться с его гибкостью на рабочем столе эпохи KDE 3.5, да и сейчас с ним следует считаться. *SuperKaramba* устанавливается по умолчанию, но при первом запуске вы не обнаружите никаких апплетов. Чтобы получить что-нибудь, нажмите на кнопку Get New Stuff вверху списка апплетов. При этом с сайта *SuperKaramba* скачается список апплетов; двойным щелчком можно выбрать из них те, что нужно установить. Затем апплет появится на вашем рабочем столе, откуда вы можете его перетащить и закрепить где-нибудь. Если у апплета есть панель настройки (например, для задания вашего местожительства при обновлении погоды), щелкните правой кнопкой по апплету, и вы увидите меню. Вы можете запускать и прекращать выполнение *SuperKaramba*, а также любых установленных апплетов, с помощью значка, появляющегося в нижней панели рабочего стола.



› **Апплеты *SuperKaramba* включают обычный набор мониторинга системы и показ погоды, а также несколько неплохих подобий Apple'овского Dock.**



› **Значки Oxygen, наверное, лучшая часть рабочего стола KDE 4, однако и для Gnome, и для KDE 3.5 существуют пакеты, способные придать тот же вид вашему любимому окружению.**



»

Compiz-Fusion

Радует глаз и совмещается с любым рабочим столом – при наличии графического ускорителя.

Compiz – одна из технологий, поразивших коллективное воображение. Некоторые хмурятся и говорят, что это трата ресурсов и пустое украшательство, но *Compiz* принес много пользы настольному Linux. Он превосходит подобные технологии для Apple OS X и Microsoft Vista, и дает не-пользователям Linux повод для разговоров. Есть хоть кто-нибудь, кто ни разу не видел 3D-куб рабочего стола, или не читал в компьютерных изданиях про колышущиеся окна?

Но главное, *Compiz* – это хорошая штука, и он уже достаточно зрел для реального применения. Последние релизы Mandriva и Ubuntu включают поддержку *Compiz* по умолчанию, а значит, вы сможете установить *Compiz* на свой любимый дистрибутив, как любое другое приложение. Это уже не мудреный процесс. Просто поищите '*Compiz*' в менеджере пакетов и установите его вместе с сопутствующими программами. Обычно можно найти пакет модулей расширения, включающий, например, кучу дополнительных эффектов. *Compiz* работает, заменяя декоратор окон вашего рабочего стола своим собственным. Декоратор окон – это процесс, ответственный за отображение окон и оконных границ. В Gnome он называется *Metacity*, а в KDE – *KWin*. Декоратор окон *Compiz* так и называется – *Compiz*, и он предоставляет собственные функции для отрисовки окон, поддерживающие 3D и композитинг.



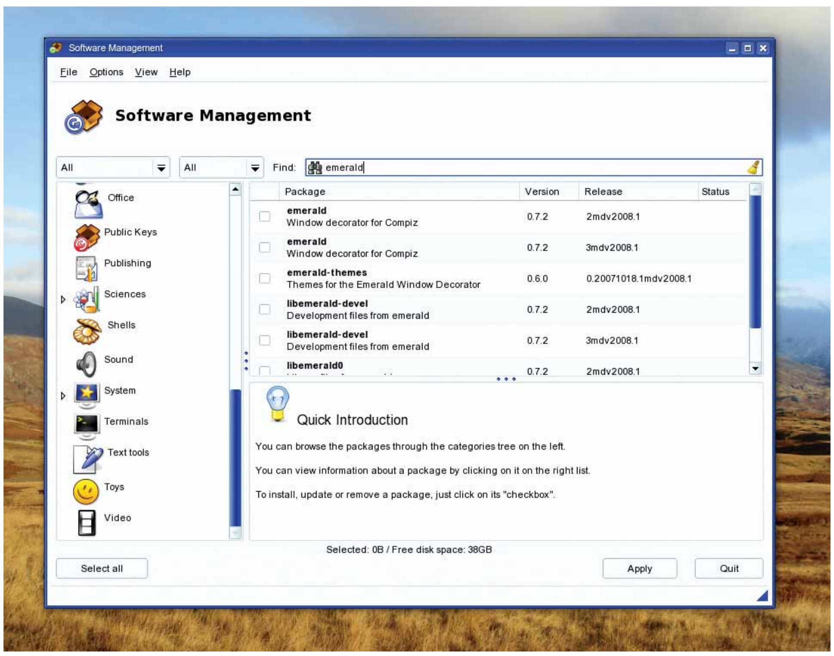
› Не все эффекты *Compiz* одинаково полезны. Например, Motion Blur скорее затрудняет пользование рабочим столом.

Установка

Кроме пакетов *Compiz*, другим обязательным требованием является наличие аппаратного графического ускорителя. *Compiz* требователен к оборудованию: старые 2D-ускорители не годятся. По сравнению с современными игровыми стандартами, требования к видеокарте весьма скромны, но 3D-ускорение необходимо. Для пользователей ATI и Nvidia это означает проприетарные драйверы. Mandriva One и Ubuntu поддерживают их установку средствами дистрибутива, для пользователей Fedora, Debian, OpenSUSE и Gentoo доступны сторонние пакеты. В случае с Nvidia, драйверы для Linux установить несложно, так как они доступны на сайте www.nvidia.com, если у вас есть установленные исходники ядра (или дистрибутив постарше). С драйверами для ATI ситуация более хитрая, но труды окупаются. После установки пакетов *Compiz* обычно есть несколько способов включить эффекты. В Ubuntu, активируйте 'Extra' в панели Visual Effects окна установок Appearance. В Mandriva, найдите путь к странице 'Configure 3D Desktop Effects' в Mandriva Control Centre и активируйте 'Compiz Fusion'. По возможности, используйте родную композиционную поддержку, а не *Xgl*, так как это более эффективно и требует меньше ручной настройки.

> «Compiz вызовет зависть у ваших друзей с другими ОС.»

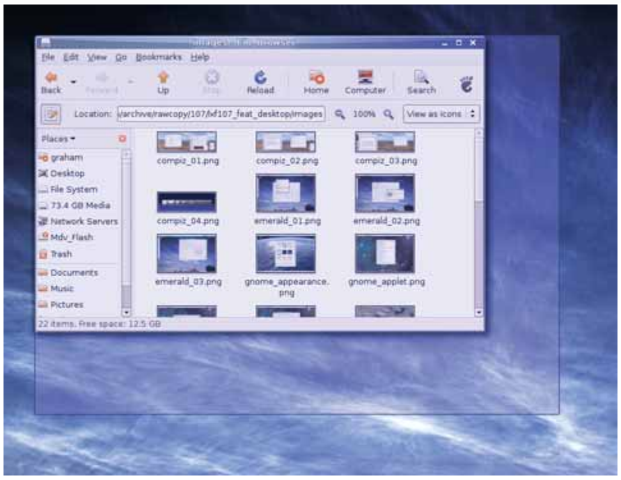
› Сейчас *Compiz* официально является частью популярных настольных систем, таких как Mandriva Spring 2008 и последний релиз Ubuntu, упростивших его установку.



Большая часть установок *Compiz* после включения выглядят весьма скромно. Обычно можно увидеть изысканные эффекты падающей тени и новый инструмент переключения задач. Это самые незатейливые эффекты. Но есть еще много чего попробовать, если вы готовы приложить к этому руки, и для этого имеется инструмент *CCSM*.

Красота – это функция

Если *CCSM* не установлен, поищите его в менеджере пакетов вашего дистрибутива либо по имени, либо по '*Compiz* Config'. Сокращение означает *Compiz Config Settings Manager*, и без него не обойтись из-за чудовищного количества опций настройки. Впервые запустив приложение, вы поймете, что мы имели в виду. В зависимости от числа установленных модулей расширения, вы, как правило, увидите десятки различных значков, каждый из которых скрывает несколько сложных и плохо документированных страниц опций. Лучший способом узнать, что делает та или иная – включить ее. Но даже тогда ситуация не проста. Например, если вам нужен куб рабочего стола, следует включить эффект Desktop Cube, а также Rotate Cube, не то вы ничего не увидите. Или вы можете решить, что включение Snapping Windows – неплохая мысль: пусть окна прилепляются к краям рабочего стола и другим окнам. Но вот беда: при попытке активировать Snapping Windows, *Compiz* сообщит вам, что сначала следует отключить Wobbly Windows.



› Модули расширения *Compiz* типа Resize Window сделают вашу работу более гладкой.

Это возмутит любителей колеблющихся окон, пока они не осознают, что можно делать и то, и другое. Snapping Windows задействуются из страницы настроек Wobbly Windows и включены по умолчанию, но работают только тогда, когда при перетаскивании окна вы держите клавишу Shift. Опция Snap inverted меняет дело, включая Snap windows всегда, когда вы перетаскиваете окно. В *Compiz* существует множество подобных вариантов и проблем, но несмотря на них, технология вызовет зависть у ваших друзей, пользующихся другими ОС. Вот некоторые из модулей расширения, достойные приложения усилий:

» **Desktop Cube** Синоним *Compiz* с самого его рождения; на каждой грани куба рабочего стола наклеен свой виртуальный рабочий стол. Увеличьте размер рабочего стола на странице General options *CCSM* – и «куб» соответственно изменится. Вам также, видимо, придется заменить ваш апплет переключателя рабочего стола на *Compiz*-совместимый, так как старые не воспринимают новый формат.

» **Resize Window** Простой эффект замещает функцию изменения размера окна оконного менеджера на нечто более изысканное. По умолчанию, это означает постоянное экранное обновление содержимого окна, но мы считаем, что этот эффект перегружает CPU. Нам кажется, что режим изменения размера «Rectangle» практичнее, так как заменяет окно прозрачным цветом – тоже приятно на вид, но требует меньше системных ресурсов.

» **Window Previews** Когда вы сворачиваете окно на панель задач, этот модуль позволяет видеть его содержимое при удерживании мыши над свернутым значком. Он также покажет предпросмотр окон на разных рабочих столах, что идеально, если у вас есть целый зоопарк приложений.

» **Enhanced Zoom Desktop** Входит в категорию Accessibility в *CCSM*, но это один из самых полезных модулей во всей установке. Он позволяет приближать область непосредственно под курсором, при удержании специальной клавиши (обычно – Windows) и прокрутке колеса мыши. Это годится для увеличения малых областей экрана, и особенно хорошо для маленьких видеоокон. LXF

Вам слово!

Мы бы хотели увидеть результаты ваших трудов над рабочим столом. Если вы пришли к чему-нибудь более-менее приличному, почему бы вам не сделать снимок экрана и черкнуть пару строк на letters@linuxformat.ru или описать ваши труды на форумах LXF? Если нам это понравится, мы, может быть, даже соберем коллекцию лучшего на LXFDVD.

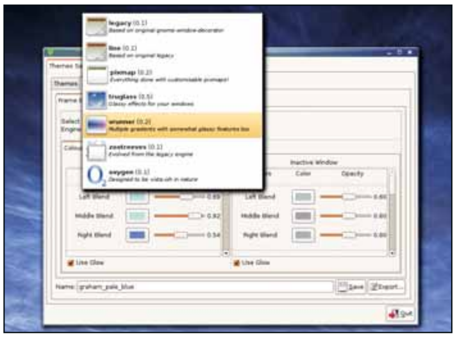


Три лучших совета: Emerald Theme Manager

Большая часть рабочих столов Linux, включая Gnome и KDE, имеют собственные оконные темы, но при переходе к *Compiz* они теряются. К счастью, у нас есть *Emerald Theme Manager*, *Compiz*-эквивалент Gnome'овским и KDE'шным утилитам, который более гибок и приятен. *Emerald* редко ставится по умолчанию, но его можно без труда найти в менеджере пакетов. Лучше всего запускать *Emerald* из так называемого апплета *Fusion-Icon*. Это небольшая утилита, позволяющая переключаться между менеджерами окон и запускать *Emerald Theme Manager* и *Compiz* из одного значка. Из *Fusion-Icon* также можно предусмотреть запуск *Emerald Window Decorator* перед запуском менеджера тем.

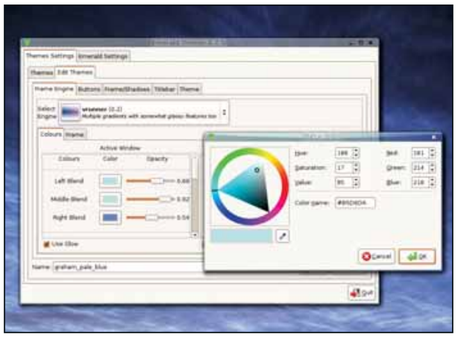
После установки *Emerald* не содержит никаких важных тем. Их можно скачать с http://themes.beryl-project.org. Для загрузки темы щелкните по кнопке Import в менеджере тем, и оформление окон рабочего стола изменится. Но для полного счастья вы можете сделать свои модификации любимой темы. С *Emerald Theme Manager* это легко: просто переключитесь на страницу Edit Theme и меняйте, что хотели.

Наибольшие различия получаются при изменении движка отображения. Он отвечает за эффект стекла, используемые многими темами для заголовка окна. Например, Oxygen схож по виду с Aero, используемый в Microsoft Vista, а Trueglass очень Apple-подобен. Другие опции относятся к раскраске рамки окна и всецело зависят от выбранного движка; и все сделанные вами изменения тут же отразятся в оформлении окна.



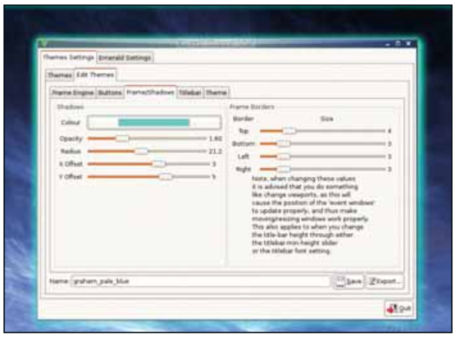
1 Движок отрисовки

Наш любимый движок называется Vrunner. Он неплох на вид и не завалит вас обилием опций.



2 Цвет заголовка окна

Мы выбрали зеленый оттенок на одной стороне и темно-синий – на другой. *Emerald* смешивает их, чтобы получилось подобие естественного отражения.



3 Падающие тени

Вы можете сделать тени любого цвета; более яркие тона дают цветущий эффект вокруг окна.

Хакеры: как все начиналось

Мир хакеров не материализовался из воздуха. **Ричард Хиллзли** разбирается, откуда он взялся, каков он сейчас и каким будет завтра.



Когда журналист Стивен Леви [Steven Levy] получил от журнала *Rolling Stone* задание разоблачить «разросшийся, враждебный и антиобщественный» мир компьютерных хакеров, он обнаружил не более чем скромное полуподпольное движение. Многие студенты, с которыми он беседовал тогда, позже стали ключевыми фигурами революции свободного ПО. Однако в то время «хакеры» считались отщепенцами, не слишком отягченными моралью, но дорвавшимися до знаний, недоступных другим. «Общественное мнение» полагало, что они затевают применить эти знания с целью вывернуть мир наизнанку… а то и хуже.

Леви обнаружил нечто совершенно иное. «Они не были асоциальными выродками: это были обаятельные люди, стремившиеся к великому. Это были художники, исследователи, первопроходцы. Они выделывали на компьютере невероятные вещи, и это восхищало меня». Книга, написанная Леви по результатам этого исследования, Hackers: Heroes of the Computer Revolution [*Хакеры: герои компьютерной революции*], была опубликована в 1984 году. Книга стала классикой оценки влиятельной субкультуры и ее вклада в развитие мира компьютеров.

От истоков

Леви отыскал корни хакерской культуры в Массачусетском технологическом институте [Massachusetts Institute of Technology, MIT]. Движение зародилось среди студентов, входивших в состав подкомитета энергетики и сигнализации (Signals and Power, S&P) клуба технического моделирования железных дорог (Tech Model Railroad Club, TMRC) в конце 50-х – начале 60-х годов.

› TX-0, впервые запущенный в 1956 году, имел целых 64 КБ памяти!

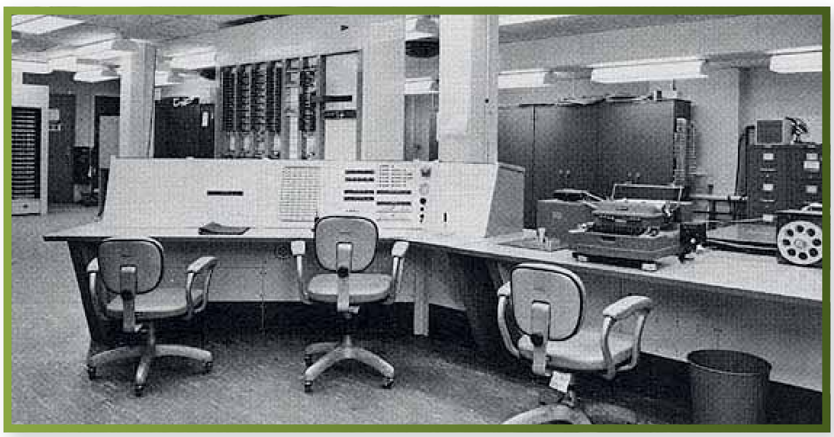


Photo: Computer History Museum

«Там-то я и наткнулся на источник всех этих компьютерных дебатов, – вспоминал Леви. – Ее [железнодорожной модели] основу составлял лабиринт переключателей и проводов, обеспечивавших управление поездами. Эти люди были первыми, назвавшими себя “хакерами” в технологическом смысле».

За моделью «стояла Система, похожая на совместное порождение Руби Голдберга [американский мультипликатор того времени] и Вернера фон Брауна», – писал Леви, – причем система постоянно улучшалась, переделывалась, оттачивалась, а иногда «падала» – на клубном жаргоне, ломалась. Люди из S&P были одержимы проблемами Системы, ее растущей сложностью; влиянием изменения одной части на все остальное; и достижением оптимального взаимодействия частей».

Полуночные бродяги

Хакеров из S&P прозвали еще «Комитетом ночных реквизиций», потому что «…если для оснащения Системы какой-либо новой функцией нужен был комплект диодов или пара добавочных реле, несколько членов S&P могли, дождавшись темноты, добыть недостающие детали там, где они имелись. Ни один из хакеров, людей как правило щепетильно-честных, не отождествлял этого с воровством».

Эти вылазки неминуемо привели к обнаружению новейшего компьютера TX-0 в корпусе № 26 института, и хакеры начали осаждать компьютер по ночам, «…когда ни один человек в здравом уме не выбрал бы час машинного времени в расписании, которое по пятницам вывешивалось рядом с кондиционером в лаборатории RLE… Хакеры TMRC, вскоре переименованные в “хакеров TX-0”, изменили стиль жизни, вобрав в нее компьютер».

Один из хакеров MIT, Боб Сондерс [Bob Saunders], позднее объяснял Леви: «Другие вовсю учились, проводя дни напролет в четырехэтажных корпусах за смрадными химическими опытами, столкновениями элементарных частиц и всяким таким. А мы просто не обращали



› **Lisp-машина теперь томится в музее MIT.**



› **Джон Маккарти, один из столпов раннего хакерского движения, создатель Lisp.**

Photo: Null0

внимания на их занятия, нам это было неинтересно. Они изучали свое, мы – свое. А то, что большая часть нашей деятельности оказалась вне рамок учебного плана, нас мало заботило».

Политические и социальные идеалы хакеров вращались вокруг машины. Ричарда Гринблатта [Richard Greenblatt], иногда называемого «хакером из хакеров», отчислили с курса: он так увлекся компьютером, что ему было не до лекций с экзаменами. Говорят, Гринблатт нечасто вспоминал о личной гигиене и отличался внешней непрезентабельностью, зато именно он написал первую шахматную программу и создал Maclisp, диалект Lisp для проекта MAC на PDP-6. Он стал также соавтором революционной «Несовместимой системы с разделением времени» (Incompatible Timesharing System, ITS), ставшей разносчиком хакерского программирования. Кроме того, он, совместно с Томом Найтом [Tom Knight], приложил руку к созданию Lisp Machine, первой коммерческой однопользовательской рабочей станции.

Первые хакеры продолжали работу: это они придумали компьютерные игры, музыкальные программы и фокусы с дисплеями. Вскоре их взяла под крыло лаборатория искусственного интеллекта (Artificial Intelligence, AI) под руководством профессоров Марвина Мински [Marvin Minsky] и Джона Маккарти [McCarthy], изобретателя Lisp. В 1963 году AI Group вошла в состав институтского проекта MAC. Аббревиатура раскрывалась двояко: Multiple Access Computing [коллективные вычисления] и Machine Aided Cognition [компьютерное познание]. Проект финансировало Агентство по перспективным исследовательским программам (Defense Advanced Research Projects Agency, DARPA) Министерства обороны США.

> «Политические и социальные идеалы хакеров вращались вокруг машины».

Маккарти и Мински всячески поощряли инициативу, пытливость и поиски сотрудников лаборатории, предоставив группе из десяти-двенадцати хакеров AI полную свободу действий, даже после расширения сферы исследований AI Group. В 1970 году Лаборатория искусственного интеллекта MIT (или AI Lab) получила самостоятельность, но исследования компьютерных технологий продолжались.

В среде хакеров TMRC и AI Lab сформировался своеобразный кодекс чести, позже известный как Моральный кодекс хакера (The Hacker Ethic). Сленг, на котором хакеры общались между собой, был запечатлен в институтском файле **jargon.txt** и стал основой «Словаря хакера». Хакеры MIT впервые определили «хак» как «быстро написанную короткую программу, обеспечивающую результат» или как «написание программ не просто ради практической цели, но и ради удовольствия от самого процесса». Они считали хакером того, кто «программи- »

› **Марвин Мински возглавлял AI Group в начале 60-х.**

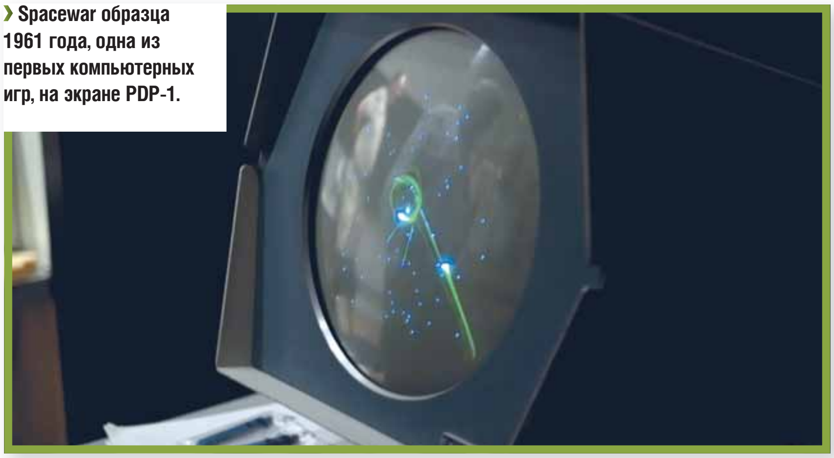


Photo: Steamtalks

Моральный кодекс хакера

Хакерская этика, в интерпретации Стивена Леви, основывается на следующих правилах:

» **Доступ к компьютерам** – как и к любым сведениям, необходимым для познания мира – должен быть неограниченным и всеобщим. Не забывай о своем праве на информацию!
» **Любая информация** должна быть свободно доступной.
» **Не верь авторитетам** – стремись к децентрализации.
» **Оценка хакера** определяется его реальными делами, а не мишурой (степень, возраст, раса или положение в обществе).
» **Твори** на компьютере чудеса.
» **Компьютер изменит** жизнь к лучшему.



› Spacewar образца 1961 года, одна из первых компьютерных игр, на экране PDP-1.

Photo: Joi Ito

» рует самозабвенно и с наслаждением, а не разводит об этом теории».

Пока Леви писал свою книгу, обстановка в AI Lab менялась, отражая изменения в окружающем мире. Благодаря гибкости и изяществу, любимым языком программирования у хакеров был Lisp, требующий много памяти. Это побудило Гринблатта и Найта, двух ключевых сотрудников AI Lab, основать отдельный проект. Lisp Machine была рабочей станцией, целиком посвященной Lisp, написанной на Lisp и предназначенной для Lisp-программистов; она объединялась в сети Lisp-машин. Мало кто тогда догадывался об этом, но Lisp-машина – в своем роде кульминация достижений хакеров – стала началом конца их образа жизни. Никто не жалел об этом так, как Ричард Столлмен, котрого Леви назвал «последним из подлинных хакеров». Для него AI Lab была всем: обществом, семьей и целью жизни.

Столлмен влился в AI Lab в 1970 году и вскоре стал одной из ее звезд: он непрерывно переписывал и совершенствовал разные части системы. В частности, он работал над редактором *TECO*, позднее переросшим в *Emacs* (от Editing MACros). Однако в 80-х растущая коммерциализация Lisp-машины компанией *Symbolics*, основанной бывшими сотрудниками AI Lab, стала соблазном для столлменовского сообщества хакеров. Леви приводит записки Столлмена:

«Больно вспоминать то время. В лаборатории остались профессора, студенты и исследователи – «не-хакеры», они не знали и знать не хотели, как обращаться с системой и оборудованием. Машины ломались и не ремонтировались: иногда их просто выбрасывали. Необходимые изменения в программном обеспечении не делались. «Не-хакеры» реагировали на это переходом на коммерческие системы, принося с ними фашизм и лицензионные соглашения. Я часто бродил по пустым ночным комнатам, которые прежде были полны народа, и думал: "Бедная AI Lab! Ты умираешь, и мне тебя не спасти…"».

Просто забава

В ответ на распад AI Lab Столлмен основал проект GNU, Фонд свободного ПО [Free Software Foundation, FSF] и разработал Универсальную общественную лицензию [General Public License, GPL], под которой ныне выпускается большинство открытых и свободных программ. Целью GNU, начатого Столлменом в 1983 году, стало создание «с нуля» абсолютно свободной операционной системы.

«Единственная причина, побуждающая нас создать целую операционную систему, – говорил Столлмен, – это убеждение в том, что операционная система должна быть свободной полностью, а не на 90%. Если не принять свободу как принцип, всегда найдется повод сделать исключение. По той или иной причине, практические соображения рано или поздно вынуждают делать исключения».

> «В ответ на распад AI Lab, Столлмен основал проект GNU.»

Девятью годами позже Линус Торвальдс объявил группе usenet comp.os.minix о предстоящем создании Linux: «Я пишу (свободную) операционную систему (просто для забавы, не такую большую и профессиональную, как GNU) для клонов 386(486) AT».

Истоки культуры, из которой выросли Linux и движение свободного ПО, восходят непосредственно к AI Lab и Моральному кодексу хакера. Особый вклад Столлмена состоит в предвидении юридических и проприетарных препятствий распространению идей свободного ПО. Язык общения на таких проектах (да и во всей ПО-индустрии) в значительной мере сформировался под влиянием трудов Столлмена. Его отношение к законам и патентам, окружающим мир ПО, можно выразить одним из принципов Морального кодекса хакера: «Доступ к компьютерам – как и любые сведения, необходимые для познания мира – должен быть неограниченным и всеобщим».

Основа замысла

Успех коллективной модели разработки программного обеспечения движением свободного ПО, прямым наследником хакерской культуры AI Lab, изменил облик всей индустрии программирования. Достижения GNU/Linux, Apache, Perl и других свободных проектов открывают перед компьютерным миром новые горизонты.

Столь разные организации, как Intel, Novell, NASA, Sun Microsystems и IBM, не только вносят идеи и публикуют ПО на условиях GPL и ее вариантов, но и активно участвуют в проектах свободного ПО к двойной выгоде: их собственных пользователей и сообщества. Их мотивы не всегда альтруистичны, но на результат это не влияет. Свободное и открытое ПО (Free/Libre and Open Source Software, FLOSS) стало стандартным способом разработки для многих компаний, и это изменило отношение к обмену и использованию информации корпорациями и индивидуальными пользователями. Компании принимают участие из эгоистичных побуждений: потому что «это работает» и дает двойной результат при вдвое меньших вложениях. Разные люди участвуют в этих проектах по разным причинам, а некоторые, как Линус Торвальдс,

Четыре свободы

Ричард Столлмен определил свободное ПО как «свободу пользователя запускать, копировать, передавать, изучать, изменять и улучшать программы». Точнее, это сводится к четырем основным свободам пользователя в отношении ПО:

0 **Свобода** запускать программы, с любой целью.

1 **Свобода** изучать принципы работы программ и приспосабливать их к своим нуждам. Доступ к исходному коду – обязательное условие.

2 **Свобода** распространять копии программ, чтобы помочь ближним.

3 **Свобода** совершенствовать программы и опубликовывать свои усовершенствования, ради общественного блага. Доступ к исходному коду – обязательное условие.



› Столлмен: а ведь он дело говорит!



Photo: Thane PLambeck

› Билл Госпер, один из первых хакеров AI Lab.

работают просто ради удовольствия. «Основа моего замысла», писал Торвальдс в 1995 году, «в том, что Linux должен приносить радость людям».

Эволюция и революция

В процессе развития в сети Интернет модель разработки ПО с открытым исходным кодом создала некоторые новые правила. Неформальность новой модели, возможность для каждого участника найти занятие по своим способностям поощряют новаторство похлеще экономических стимулов. Очень важно, что весь процесс разработки подвержен равноправному обсуждению коллег-хакеров. Чтобы получить стоящий продукт, необходим талантливый замысел, а воплощение талантливого замысла следует подкрепить критикой коллег – принцип, общепризнанный в науке. Процесс развития человечества основывается на общей совокупности знаний (нечто подобное обеспечивает открытый исходный код программы), и в новых исследованиях используется весь багаж сведений, накопленный прежними человеческими поколениями. В этом смысле разработка программ с открытым исходным кодом – лишь вариант традиционного процесса.

Моральный кодекс хакера в значительной мере сформировал культуру раннего Интернета. Тим Бернерс-Ли [Tim Berners-Lee] в своей книге *The World Wide Web: A Very Short Personal History* [*WWW: Краткая история*] писал: «Идея всемирной паутины – единое пространство, где все мы общаемся, обмениваясь информацией. Пространство всеобъемлющее и вездесущее, ведь гипертекстовая ссылка может указать на что угодно, будь то личное, местное или глобальное; набросок, эскиз или нечто завершенное. Есть и вторая часть идеи: всеобщее использование Интернета привело к тому, что он стал реалистичным зеркалом (или даже первичным воплощением) нашей работы, нашего досуга и нашего взаимного общения. Ну, а коль скоро наше взаимодействие переместилось в Сеть, мы смогли при помощи компьютера проанализировать состояние вещей, оценить свою деятельность, определить свое место во всем этом и продумать, как лучше объединить усилия для совместной работы».

С коммерческой точки зрения разработка ПО с открытым исходным кодом – это революция, ведь в бизнесе впервые поняли, какие преимущества дает объединение усилий и обобществление накопленных знаний. Ошеломляющие результаты получены при опробовании такой модели в создании анимации, спецэффектов и биотехнологиях.

Существование свободного ПО сломало входные барьеры: здесь можно начать с малого, просто вливаясь в существующий проект, обеспечивая ему поддержку и добавляя ценности; или начать новый проект, требующий привлечения крупных ресурсов. Еще одно достоинство подобных методов работы – они открывают доступ к технологии тем, кто иначе был бы отрезан от мирового развития. Джордж Бернард Шоу высказался однажды: «Пусть у тебя и у меня есть по яблоку. Если мы обменяемся яблоками, у каждого из нас останется по одному. Но если у тебя есть идея и у меня есть идея, и мы обменяемся идеями, у каждого из нас будет уже по две идеи».

Свобода информации

Принцип «Информация должна быть свободной» затронул не только развитие свободного ПО и Интернета: он резко контрастирует с засильем патентов и авторских прав, удерживающих знания и власть в руках корпораций. Ценность свободы и сотрудничества, обеспечиваемых GPL, коллективные методы работы, которые она поощряет, открыли миру ПО безграничные возможности, неочевидные на момент создания этой лицензии.

«Моральный кодекс хакера сформировал культуру раннего Интернета.»

Модель разработки FLOSS (и доступность продвигаемых ею идей) весьма привлекательна для широкого разнообразия проектов, в том числе и весьма далеких от специфичного мира компьютерных программ. LXF

› Стивен Леви – человек, первым проникший в мир хакеров.



Photo: Cory Doctorow





По рецептам д-ра Брауна

Добротное администрирование систем из причудливых заворотов кишок серверной.

Д-р Крис Браун
Доктор проводит обучение, разработки и консультации по Linux. Ученая степень в физике элементарных частиц ему в этом совсем не помогает.

Цапля вылупилась

Вы, наверное, прочитали обзор Ubuntu 8.04 LTS на стр. 12. Это важный релиз для Canonical, так как LTS (“long term support”) означает, что они обеспечивают поддержку в течение трех лет для настольных систем и пяти лет для серверных.

Предыдущий релиз с меткой ‘LTS’, Ubuntu 6.06, увидел свет два года назад. Новый релиз продолжает сформировавшуюся алфавитную последовательность дурацких аллитеративных названий и носит имя Hardy Heron [Стойкая Цапля].

Стойким админам

В Hardy Heron есть кое-что интересное для системных администраторов. X.Org 7.3 обещает лучшую автоматическую настройку и новую утилиту для задания разрешения и частоты обновления экрана для внешних мониторов и проекторов. (Самому интересно, как это работает: когда-то я натерпелся горя, пытаясь управлять внешними проекторами с ноутбуков с Linux.)

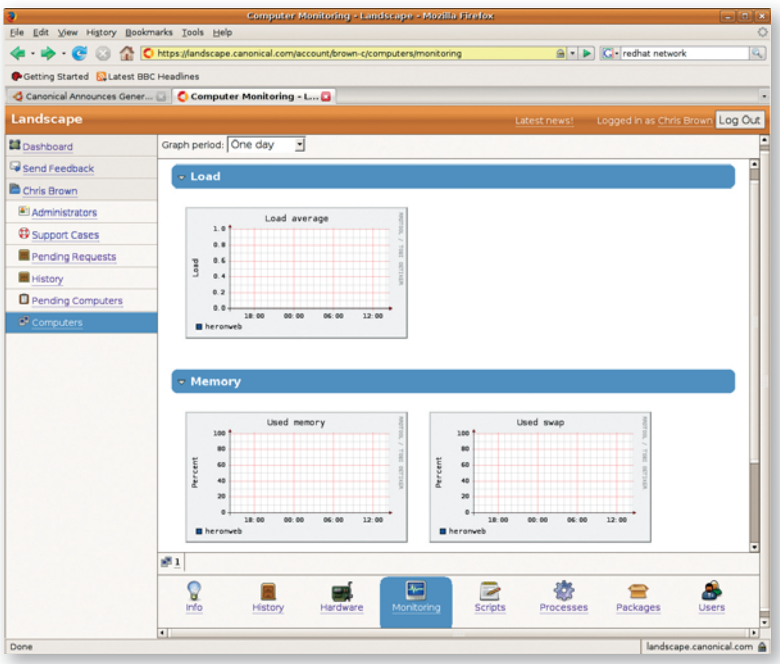
В Hardy Heron есть ожидаемые обновления ядра и свежие версии основных программ, но самое интригующее дополнение называется *PolicyKit*. Это каркас для реализации тонкой «эскалации привилегий», которая облегчает жизнь пользователям, выполняющим специфические административные задачи. Если я правильно понял, *PolicyKit* должна заменить *sudo*. Есть *Likewise Open*, приложение с открытым исходным кодом, которое поддерживает аутентификацию пользователя в Linux в Active Directory сети Microsoft. Виртуализация поддерживается через *KVM*. Для упрощения настройки централизованного брандмауэра есть новый инструмент *UFW* (uncomplicated firewall – простой брандмауэр).



Получите общую картину

Landscape Управление системы и ее мониторинг – теперь в удобном пакете webmin.

Недавно Canonical анонсировала выпуск *Landscape* – web-приложения Ubuntu для управления системой и ее мониторинга. Оно должно составить конкуренцию Red Hat Network и является еще одним подтверждением того, что Canonical серьезно намерена войти в рынок серверных и промышленных систем. *Landscape* распространяется по подписке, но на сайте www.canonical.com/landscape/register можно бесплатно зарегистрироваться на 60-дневный пробный период, если у вас пять и более компьютеров. Сразу после подписки нужно загрузить клиентские пакеты *Landscape* на каждый компьютер, которым нужно управлять, затем зарегистрировать их в сервисе. Зарегистрированные клиенты регулярно отсылают на сервер *Landscape* отчеты о своем состоянии. На рисунке показан один из доступных отчетов с загрузкой процессора и памяти. Прямо из браузера можно запустить произвольные скрипты на любом количестве компьютеров, также есть окно для просмотра и управления запросами службы поддержки Canonical.



› *Landscape* осуществляет удаленный мониторинг загрузки процессора и использования памяти, списка оборудования и ПО, процессов и пользовательских учетных записей.

Магия SysRq

Как быстро привлечь внимание ядра, или почему Так Невообразимо Нудно Растить Слонов.

В следующий раз, когда окажетесь неподалеку от Linux-компьютера своего друга, проведите следующий эксперимент: нажмите Alt и SysRq и затем B. Потом с удивлением отойдите назад и скажите: «Странно – твой компьютер, кажется, перезагрузился». На самом деле, вы только что наблюдали чудо Волшебной Клавиши SysRq.

Название Magic SysRq всегда будило во мне фантазию, что это предмет поиска для Бильбо Бэггинса [Bilbo Baggins – хоббит, персонаж произведений Джона Р. Р. Толкиена, – прим. пер.] в какой-нибудь пещере. На самом же деле это комбинации горячих клавиш для быстрого привлечения внимания ядра.

В таблице показаны некоторые команды, доступные с помощью Magic SysRq. Помните: чтобы это работало, нужно одновременно нажать Alt и SysRq. Если хотите поэкспериментировать, предупреждаю, что реакция ядра на эти клавиши очень резкая. Если у вас есть несохраненные файлы, не выполняйте этих команд! Также, возможно, придется поэкспериментировать, чтобы подобрать правильную комбинацию. Например, у меня клавиатура Microsoft с клавишей, помеченной как F Lock. Чтобы клавиша SysRq заработала, ее нужно отключить. На ноутбуке должна быть клавиша Fn – чтобы заработала клавиша SysRq, ее, возможно, придется нажать, так что в итоге получается комбинация Fn+Alt+SysRq+B.

Да, но зачем?

Вы можете спросить, зачем все это нужно. Согласен: это, наверное, не то, что требуется каждый день. Я как-то использовал Magic SysRq для имитации ошибки в системе (для проверки сбоя отказоустойчивого кластера). Не сомневаюсь, что гуру ядра смогут выудить полезную информацию из списка задач и статистики использования памяти, которые доступны по клавишам T и M.

Если система реально подвисла, можно перезагрузить ее более элегантно, чем просто отключив питание. Рекомендованная последовательность клавиш R > E > I > S > U > B (переключение клавиатуры из raw-режима, отправка сигнала SIGTERM всем процессам в надежде, что они завершатся сами, отправка сигнала SIGKILL тем, кто этого не сделал, сброс всех смонтированных файловых систем на диск, перемонтирование их в режиме «только для чтения» и перезагрузка). Чтобы запомнить эту последовательность, пригодится акроним “raising elephants is so utterly boring” (Растить слонов ну просто невообразимо нудно); можно запомнить это как ‘busier’ наоборот. Для лучшего результата делайте паузу в несколько секунд между нажатиями клавиш.

Если вам не удается заставить Magic SysRq работать, нужно кое-что проверить. Во-первых, эта возможность должна быть включена в ядре. Для этого просмотрите копию файла конфигурации ядра, которая (будем надеяться) лежит в каталоге **/boot**. Файл, скорее всего, называется как-то вроде **config-2.6.18-Нечто**, и строка, которую мы ищем, определяет параметр CONFIG_MAGIC_SYSRQ. Если он установлен в ‘y’, все хорошо, если нет – потребуется изменить конфигурацию ядра и перекомпилировать его.



Параметры ядра

Во-вторых, параметр времени выполнения ядра **/proc/sys/kernel/sysrq** нужно установить в **1**. Эта установка времени выполнения, а не времени сборки, и ее можно установить динамически с помощью команды:

```
echo 1 > /proc/sys/kernel/sysrq
```

или

```
sysctl -w kernel.sysrq=1
```

Если вы хотите, чтобы изменение стало постоянным (т.е. сохранилось после перезагрузки), добавьте строку

```
kernel.sysrq=1
```

в файл **/etc/sysctl.conf**.

Я только что проверил все системы, которые у меня под рукой: в Ubuntu 7.04, 7.10 и 8.04 функция Magic SysRq включена по умолчанию. В Fedora 7 она настроена в ядре, но отключена в **/proc/sys** (лично мне кажется, что это более безопасный вариант).

Держим Alt + SysRq + клавишу...

Клавиша	Действие
0-9	Установить уровень журналирования, что влияет, например, на вывод, генерируемый командами M и P.
R	Переключить клавиатуру в режим XLATE.
E	Отправить сигнал SIGTERM всем процессам, кроме *init*. Ожидается, что приложения ответят на этот сигнал, приведут все в порядок и корректно завершатся.
I	Отправить сигнал SIGKILL всем процессам, кроме *init*. Это принудительно завершает все процессы.
S	Синхронизировать все файловые системы (т.е. записать буферизованные данные на жесткий диск).
U	Перемонтировать все файловые системы в режиме «только для чтения».
B	Незамедлительно перезагрузиться.
O	Отключить питание.
M	Вывести статистику использования памяти в консоль.
T	Вывести список задач в консоль.
H	Показать краткую информацию по использованию всех клавиш SysRq.

А вот налог пришел

На днях ко мне зашел налоговый инспектор. Приятный парень. В рассказе о том, чем я зарабатываю на жизнь, неизбежно был упомянут Linux. Я спросил, слышал ли он о нем. «О да», – сказал он, и я был сильно польщен и ободрен. Но потом он немного подпортил впечатление, сказав: «Это язык программирования, да?»



Назад, к основам

Первое правило системного администратора гласит «никому не говори о системном администрировании». Второе – «резервируйте как маньяки».

Диски сейчас достаточно надежны, и легко поддаться иллюзии, что хранимые на них данные вечны. Чтобы открыть нам глаза на их эфемерность, нужна катастрофа. Есть такая шутка, что в мире существует два типа системных администраторов – те, кто регулярно делают резервные копии данных и те, кто хотел их сделать. За последние десять лет лично у меня было два сбоя жесткого диска. Последний случай был с нашим Sky HD, и он, наверное, не в счет. Второй – с ноутбуком. Но я слышал и о других, включая незабываемую историю, когда ребенок решил отключить папин USB-диск и спустил его в унитаз. Здорово.

За годы я перепробовал многие способы резервного копирования. В своей прошлой жизни (до Linux), я пользовался двумя программами: *dump* и *restore*. Мы записывали дамп на полудюймовую магнитную ленту 800bpi (бит на дюйм) – это, наверное, было где-то в 1977. Важно отметить, что *dump* и restore работали с инкрементными резервными копиями. Мы начинали с дампа «уровня 0», в котором было все. На следующий день делался дамп «уровня 1», который содержал лишь изменения по сравнению с предыдущим уровнем 0, и так далее. Инкрементальное резервное копирование – очень выгодная штука, потому что большинство файлов не изменяется за день и можно сэкономить массу времени, полосу пропускания сети и носители.

> «За последние десять лет лично у меня было два сбоя жесткого диска.»

Живое ископаемое

Удивительно, но эти две древние программы дожили до сегодняшнего Linux (см. dump.sourceforge.net). *dump* и *restore* работают не как большинство программ. Они не получают доступ к данным файла через обычные системные вызовы, а открывают устройство раздела диска (например, **/dev/sda1**) и взаимодействуют со структурой файловой системы напрямую. Отсюда – несколько важных следствий. Во-первых, *dump* и *restore* работают только с файловыми системами ext2 и ext3. Они не могут, например, создать резервную копию образов файловых систем Reiser или FAT32. Во-вторых, желательно не пользоваться ими на рабочих файловых системах, во всяком случае, не на критических. Некоторые администраторы перед резервным копированием размонтируют раздел, другие монтируют его в режиме «только для чтения». На интенсивно работающих серверах предприятия ни один из этих подходов применять нельзя.

Третий подход, для случая, когда вы используете логические тома, это снять образ диска и сделать его резервную копию. Четвертый подход – махнуть на все рукой и надеяться на лучшее. Так как эти ограничения на предприятиях очень досаждают, *dump* и restore почти полностью утратили популярность. Однако есть и одно довольно тонкое преимущество работы ниже слоя виртуальной файловой системы – резервное копирование файловой системы не влияет на временные метки файлов.

Более популярный, по крайней мере для небольших и средних систем, способ резервного копирования – просто поддерживать с помощью *Rsync* актуальную копию файловой системы, скажем, на сменном жестком диске, на сервере для резервного копирования в сети или даже – если скорость позволяет – на диске, поставляемом вашим провайдером. Этот подход подробно описан Джульеттой Кемп [Juliet Kemp] в ее превосходном руководстве по *Rsync* в LXF105, и я не буду здесь его обсуждать.

Есть и третий инструмент, очень популярный для создания и хранения архивов – *tar*. Например, многие пакеты на сайте SourceForge хранятся в виде *tar*-архивов. В *tar* удобно то, что архивы можно сжимать. Навскидку: сжатый *tar*-архив каталога **/lib** на моем компьютере имеет размер 90 МБ; сама файловая система занимает 262 МБ. Второй тест – архивирование 400 МБ картинок в JPEG – дает выигрыш менее одного процента, что неудивительно, так как эти файлы сжаты сами по себе.

Резервное копирование с tar

Хотя *tar* широко используется для архивирования данных, он редко применяется для ежедневного резервного копирования, так как не может создавать инкрементные резервные копии – по крайней мере, так считает большинство. На самом деле в версии GNU есть отличный механизм создания и восстановления инкрементных архивов, он просто не очень хорошо документирован на man-странице, и описание приходится искать на сайте GNU. В его основе – сохранение дополнительных метаданных в отдельном файле, файле снимка. Проиллюстрируем это на небольшом примере.

Предположим, я начинаю в «первый день» с каталога **mypics**, в котором есть три файла:

```
$ ls ~/mypics
caption.jpg storm1.jpg sunset1.jpg
```

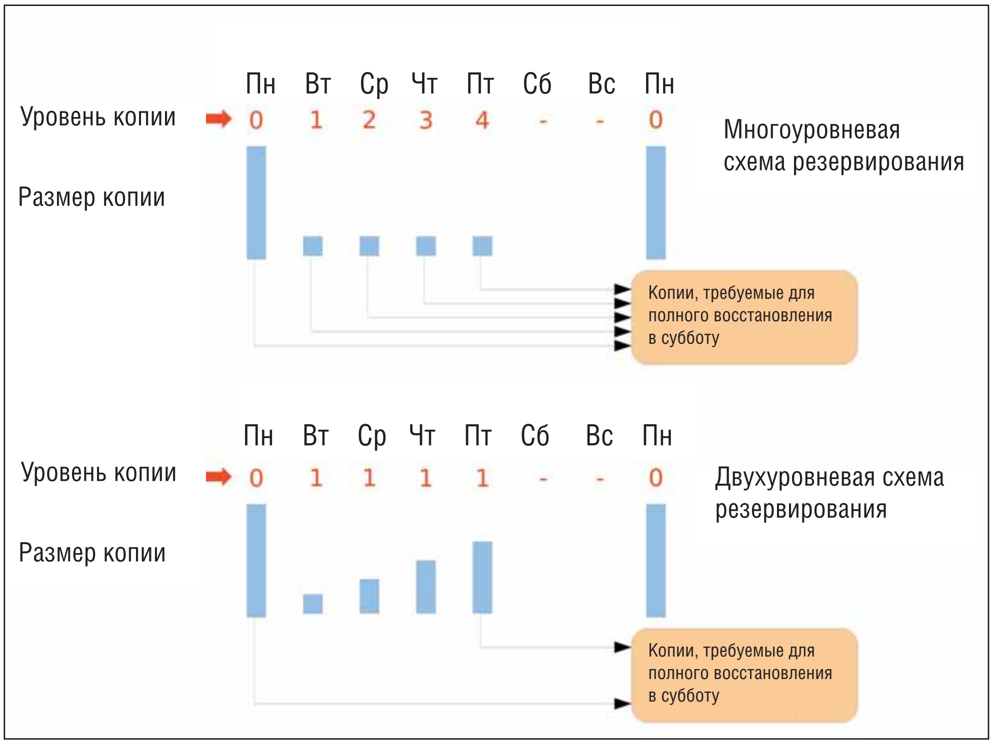
Если создать архив *tar*, в нем будут все три файла. Это наша резервная копия «уровня 0»:

```
$ cd ~/mypics
$ tar cvf /backups/mypics.0.tar -g mypics.snar .
./
./caption.jpg
./mypics.snar
./storm1.jpg
./sunset1.jpg
```

Первый аргумент *tar* (cvf) – это на самом деле набор из трех опций.

6 постулатов резервирования

1. Самое важное – не выбрать самую последнюю, самую быструю технологию с суперсжатием, а гарантировать, что вы будете его делать, каким-то разумным способом, на последовательной и регулярной основе. Резервное копирование чем-то похоже на уплату страховых взносов: вы надеетесь, что страховой случай никогда не наступит, и появляется соблазн не платить их совсем.
2. Создавать резервные копии файловой системы на том же жестком диске – все равно, что назначать свидание Карле Саркози, т.е. пустая трата времени. Не делайте этого.
3. Если вы сохраняете резервную копию на другой компьютер в сети, помните, что если взломают ваш компьютер, то и до сервера резервных копий тоже могут добраться. (Ничто не убедит вас в сохранности данных сильнее, чем три метра расстояния между локальной сетью и внешним USB-накопителем на полке.)
4. Если резервные копии сохраняются на сменном диске, помечайте его!
5. Храните внешние носители (CD или жесткие диски) в других помещениях. Мне помогают в этом соседи. Конечно, они получают доступ к личным данным, поэтому им нужно доверять (или предполагать, что они не сумеют добраться до данных).
6. Какой бы метод вы ни использовали, убедитесь, что данные на деле можно восстановить. Проведите «учения» – представьте, что несколько файлов утеряны, и восстановите их.



› **Двухуровневая и многоуровневая схемы резервного копирования предоставляют компромисс между размером резервных копий и сложностью процесса полного восстановления.**

с означает «создать архив», v – «подробный» (выводить имена файлов во время их записи в архив), а f – «следующий аргумент будет именем файла-архива». Здесь я предполагаю, что /backups – точка монтирования файловой системы сервера NFS или внешнего диска. Интересен флаг -g. Он велит *tar* сохранить запись о том, что было архивировано (и когда) в файл снимка **mypics.snar**. Наконец, вроде бы ничего не значащая точка . в конце команды – это имя каталога, который нужно архивировать, в данном случае – текущая директория.

На второй день я добавил в каталог новый файл **baby.jpg**. Я создаю другой архив. Он содержит только новый файл и является нашим «уровнем 1»:

```
$ tar cvf /backups/mypics.1.tar -g mypics.snar .
./
./baby.jpg
```

На третий день я могу продолжить, создав резервную копию уровня 2 следующим образом:

```
$ tar cvf /backups/mypics.2.tar -g mypics.snar .
```

Следует понимать, что цифры, которые я добавил в выходные файлы, исключительно для меня, и уровень архива они не контролируют. Он обрабатывается файлом снимка **mypics.snar**. Пока один и тот же файл снимка продолжает обновляться, каждый архив будет инкрементным по отношению к предыдущему.

Итак, предположим, что по какой-то причине мы потеряли содержимое каталога **mypics**, и нужно восстановить его из резервной копии. Потребуется разархивировать каждый уровень по порядку:

```
$ tar xvf /backups/mypics.0.tar -g /dev/null
./
./caption.jpg
./mypics.snar
./storm1.jpg
./sunset1.jpg
$ tar xvf /backups/mypics.1.tar -g /dev/null
./
./baby.jpg
$ tar xvf /backups/mypics.2.tar -g /dev/null
./
./chris1.jpg
```

Во время восстановления нам все еще нужен флаг -g, чтобы обеспечить инкрементное поведение, но в этом случае файл снимка фактически не требуется. Обычно в таких ситуациях указывается **/dev/null**, но сойдет и все, что угодно. При извлечении из инкрементной резервной копии *tar* пытается восстановить точное состояние файловой системы, которое было при создании архива. В частности, будут удалены файлы, которые не существовали в своих каталогах на тот момент.

При архивировании по приведенной выше схеме каждый день создается резервная копия нового уровня. Альтернативная схема может быть такой: начинаем с архива уровня 0, затем каждый день создаем только архив уровня 1. Конечно, этот архив мало-помалу разрастается, но зато из него проще восстановить данные, так как нужен лишь архив нулевого уровня и последняя версия архива уровня 1. Это потребует некоторой ручной настройки файла снимка. В частности, будет нужно создать рабочую копию этого файла для создания резервной копии уровня 1 на второй день, и для создания следующей версии уровня 1 на третий день тоже нужна рабочая копия исходного файла. Во второй день вы делаете примерно следующее:

```
$ cp mypics.snar mypics.snar-2
$ tar cvf /backups/mypics.day2.1.tar -g mypics.snar-2 .
```

и на третий день делаете то же самое снова:

```
$ cp mypics.snar mypics.snar-3
$ tar cvf /backups/mypics.day3.1.tar -g mypics.snar-3 .
```

Сохраним на века

Сколько, вы думаете, проживет носитель с резервной копией? Месяц? Год? Десять лет? Век? Ответ на этот вопрос может сильно повлиять на выбор технологии резервного копирования. В восьмидесятых я встречал совершенно безумных администраторов, которые бережно хранили резервные копии своих баз данных на полудюймовой магнитной ленте, хотя у них больше не было компьютера с соответствующей лентопротяжкой. Наверное, то же самое сейчас происходит с дискетами. Исходный код тех важных утилит, который вы бережно записали на восьмидюймовую дискету в 1977 году – сколько человек сможет восстановить их сейчас? И, если вы уже улыбнулись, сколько времени пройдет, прежде чем автор статьи напишет в журнале то же самое «об этих древних CD и DVD» и пошутит про «смешные старые USB-брелки всего на 4 ГБ»?



Что за штука... Akonadi

Где найти тот электронный адрес? А какой номер телефона у босса? Когда следующая летучка по проекту? **Марко Фиоретти** объяснит, откуда *Akonadi* знает все это, и не только…

» *Akonadi* – это очередное KDE-название, тупо выбранное из-за наличия буквы K?

А вот и нет! *Akonadi* – имя пророческой богини справедливости у народов Ганы, но выбрали его просто потому, что другие имена были заняты. Вот и все!

» И что это такое?

Новая унифицированная служба управления данными PIM для KDE 4.

» Ну, а PIM что означает?

Personal Information Management, управление личными данными, то есть всеми сведениями, которые либо возникают в результате ваших связей с окружающими, либо необходимы для поддержания этих связей и контроля за вашей ежедневной деятельностью: от сообщений электронной почты до адресных книг, календарей, расписаний событий и планов будущих акций. «Управлять» всем этим значит не только поддерживать упорядоченные, организованные всеобъемлющие архивы. Обычно PIM еще включает всякие полуавтоматические функции вроде управления псевдонимами, предупреждения накладок при назначении встреч сотрудников – да хоть бы и окно, напоминающее о дне рождения вашей жены!

> «Akonadi обеспечивает чтение/запись, а также поиск любых PIM-данных»

» Но ведь в KDE это уже есть: я сам ежедневно пользуюсь *KMail* и *Kontact*!

Да, но PIM-компоненты KDE 3 – это самостоятельные программы с независимыми, подчас дублирующими друг друга механизмами управления данными и их хранения. Это создает некоторые проблемы, решить которые и призван *Akonadi*.

» А какие именно проблемы?

Первая – производительность: чтобы отобразить календарь, *KOrganizer* загружает все зарегистрированные события, а *KAddressBook* выводит все почтовые адреса, тогда как нам нужны только некоторые. С персональным компьютером это не проблема, но вообразите, что выдаст сервер коллективной работы, где хранятся данные обо всех 10 000 сотрудников корпорации! Здесь скорость падает вполне ощутимо: частью из-за избыточности архитектуры, частью из-за дублирования кода отдельных библиотек.

» Еще проблемы есть?

Недостаточная поддержка асинхронного доступа от разных приложений и уведомлений об изменении данных. Первое неудобство может привести к порче данных при попытке изменить их двумя приложениями одновременно. Вторая позиция означает, что при обновлении адреса в адресной книге почтовая программа «не замечает» новой записи до тех пор, пока не перезапустится.

» Как же *Akonadi* устранит эти проблемы?

Программа обеспечит единый подход к чтению, поиску и записи всех видов PIM-данных и метаданных за счет архитектуры, изначально рассчитанной на поддержку совместного доступа, своевременных уведомлений и мощных поисковых механизмов. Несмотря на все это, *Akonadi* – не сервер: это своего рода банк данных, централизованный кэш с унифицированным асинхронным интерфейсом для доступа. Реальные данные хранятся в отведенных им в файловой системе местах, в неизменных форматах (Maildir для почты или iCal для календаря).

» И как это решает вышеописанные проблемы?

Кэш единый, и в память загружается только по одному экземпляру всех PIM-данных, а сделанные изменения моментально становятся доступны всем программам-клиентам. Кроме того, наличие единого интерфейса для создания запросов разных типов позволяет приложениям рабочего стола запрашивать только те данные, которые им необходимы, что повышает производительность.

А еще, единый асинхронный интерфейс позволяет получать данные, не вмешиваясь в деятельность программ-клиентов.

» Я домашний пользователь, и оперативной памяти на моем компьютере полно. Нужна ли мне *Akonadi*?

Если вы интенсивно пользуетесь *KMail* и *Kontact* и у вас много друзей – вероятно, да. *Akonadi* может существенно снизить потребление памяти, а это никогда не помешает. Кроме того, *Akonadi* способен упростить решение других задач, от синхронизации адресных книг и календарей с удаленными серверами до интеграции PIM-системы с интеллектуальными поисковыми механизмами рабочего стола.

» Ладно, давайте выясним, как *Akonadi* работает. Что там внутри?

Akonadi состоит из четырех основных компонентов: отдельного хранилища для всех PIM-данных и связанного с ним кэша; менеджера уведомлений; элементов поискового механизма и библиотеки API (Application Programming Interface, интерфейса прикладной программы). Каждый компонент – это отдельный процесс, который в случае зависания или сбоя может быть перезапущен независимо от других.

» Могут ли другие приложения получать доступ к банку данных не через *Akonadi*?

Строго говоря, да: ведь данные хранятся в стандартных форматах, и к ним по-прежнему будет открыт доступ самых разнообразных программ. Но успех не гарантируется – по крайней мере, первые версии *Akonadi* могут не справиться с такими вторжениями. Еще в *Akonadi* есть менеджер уведомлений – он немедленно сообщает всем связанным программам о добавлении, удалении и изменении любого PIM-объекта, хранящегося в кэше – и поисковые программы, это внешние утилиты, фильтрующие PIM-данные по запросам других приложений.

» Есть еще что-нибудь интересное?

Да: можно использовать так называемые «агенты» и «ресурсы». Первые – это модули, обрабатывающие данные в хранилище *Akonadi*, а ресурсы – модули, подключающие это самое хранилище к внешним источникам данных, например, серверам коллективной работы (groupware).

Очень уж сложно звучит. Как же я, не будучи программистом, смогу проверить исправность *Akonadi*?
Akonadi снабжается утилитой системного лотка, с помощью которой можно будет, по крайней мере, остановить или запустить программу, а также проверить ее состояние. При необходимости эта утилита сможет выдать сообщение об ошибке от *Akonadi*.

Стоп, мы совсем забыли о библиотеке!
Точно! Именно библиотека делает возможным все это: позволяет *Akonadi* взаимодействовать с другими программами на языке сравнительно высокого уровня (в стиле *Qt*-библиотек), независимом от платформы.

Библиотека, вполне предсказуемо называемая *libakonadi*, может использоваться как приложениями рабочего стола (*Kontact*, *KOffice* или *Evolution*), так и другими службами, направляющими данные в *Akonadi* и использующими их.

***Akonadi* «признает» только KDE?**
Теоретически, нет: этим может пользоваться любая программа. *Akonadi* предлагается, в частности, как замена Evolution Data Server (EDS). На практике, хотя основные компоненты не зависят ни от KDE, ни от *Qt*, для *libakonadi* библиотеки *Qt* необходимы. Эта зависимость, а также зависимость от сервера баз данных, снизит популярность *Akonadi* за пределами KDE.

Какие преимущества могут получить (не-KDE) программисты от использования *Akonadi*?
Akonadi предоставляет улучшенные средства запроса, отображения и изменения PIM-данных для любой программы, которой они могут понадобиться. Это существенно упрощает написание прикладных программ. Например, разработчикам *Mailody* – одного из почтовых клиентов KDE – благодаря наличию *Akonadi* хватило десяти минут для написания простейшего просмотрщика почты (см. www.omat.nl/drupal/creating-mail-reader-10-minutes).

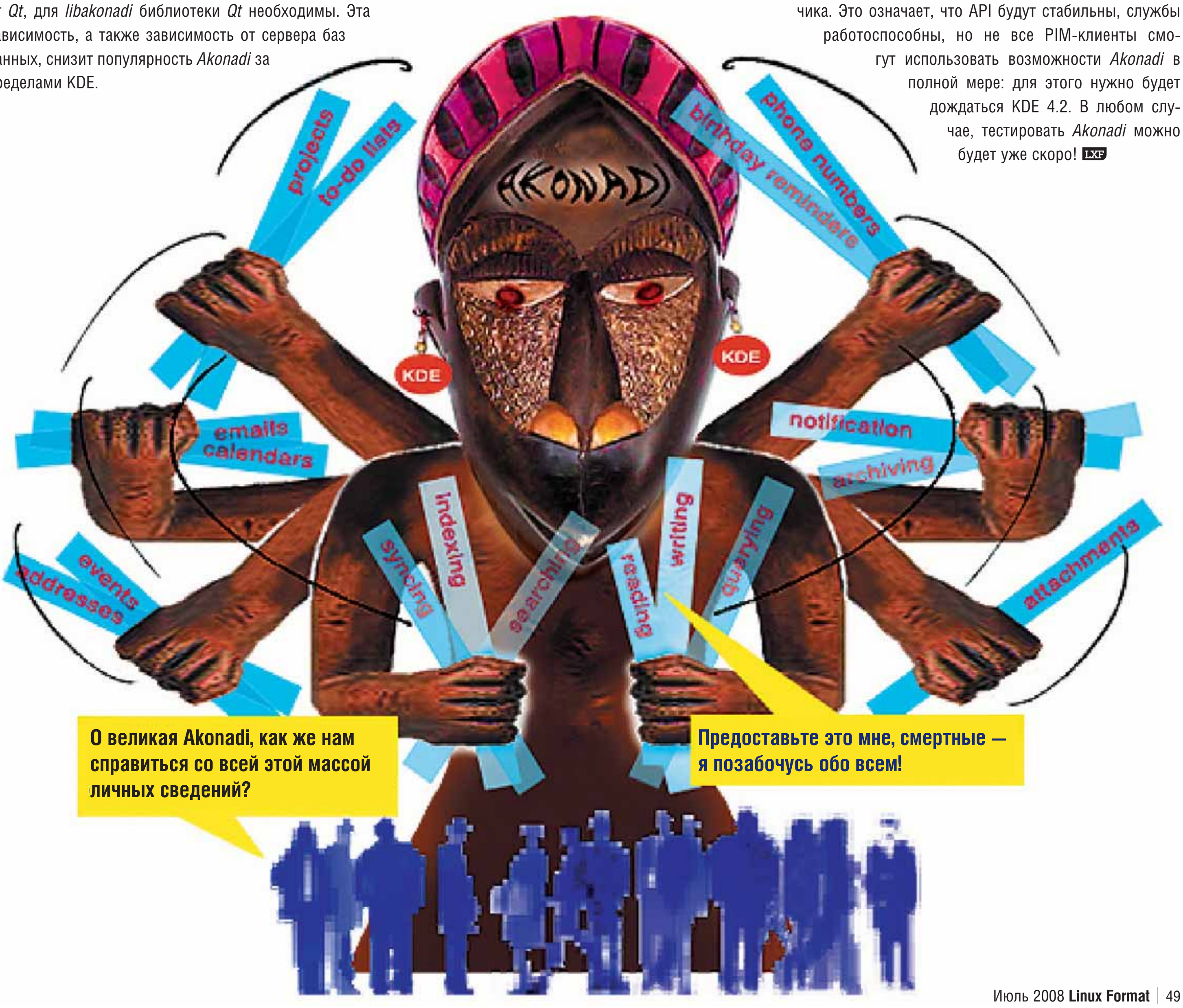
Вы упомянули, что *Akonadi* можно интегрировать с интеллектуальными поисковыми программами для рабочего стола...
Да, это конечная цель. Чтобы достичь ее, *Akonadi* потребуется координация с *Nepomuk* (http://nepomuk.kde.org), KDE-версией семантического рабочего стола. На первом уровне интеграции состоится замена категорий классификации данных, применяемых нынешними PIM-программами KDE, на тэги *Nepomuk*. Затем можно будет создать «агенты», которые будут автоматически поставлять данные из *Akonadi* в *Nepomuk* для должной индексации.

Шквал кодирования приведет к созданию таких вещей, как виртуальные почтовые папки, куда будут поступать результаты поисков *Nepomuk*. Например, можно будет запросить связанную с *Akonadi* почтовую программу на выдачу всех сообщений с рисунками во вложениях, или сообщений от тех, кто участвует в совещаниях по теме 'KDE-PIM', или даже задать сложный запрос на выдачу сообщений, более или менее близко связанных с указанной темой.

Возможна и еще одна выгода от связки *Akonadi* и *Nepomuk*: интеграция с другими, «не-PIM» приложениями. Вообразите, например, отображение фотографий и телефонных номеров своих коллег рядом с их домашними директориями в файловом менеджере…

Чуть не забыл: раз уж мы говорим о почте, календарях и адресных книгах: будет ли *Akonadi* совместим с *Exchange*?
Сейчас ведутся работы по обеспечению совместимости с *MS Exchange* при помощи библиотеки OpenChange MAPI (www.openchange.org). Но это отдельный проект, его целью является применение протоколов *Microsoft Exchange* программами открытого ПО в среде Unix и Linux.

Когда же все будет готово?
KDE 4.1, выход которого ожидается в июле 2008, включит *Akonadi* как платформу для разработчика. Это означает, что API будут стабильны, службы работоспособны, но не все PIM-клиенты смогут использовать возможности *Akonadi* в полной мере: для этого нужно будет дождаться KDE 4.2. В любом случае, тестировать *Akonadi* можно будет уже скоро! LXF





» **Кто**
Дмитрий Павлович Макаров, начальник отдела автоматизации компании «АйТи Парус».

» **Откуда**
Город Ростов-на-Дону.

» **Проект**
Перевод компьютерного парка компания «Мидитекс Инвест» на Gentoo Linux.

» **Результат**
Уже год «полет нормальный». Связка *1С+WINE@Etersoft* экспериментальным путем доказала свою жизнеспособность. Заказчик сэкономил около $60000.

» Рубрику ведет Евгений Балдин





Экспериментальная проверка GNU/Linux в Ростове-на-Дону

Как перевести компанию более чем с пятьюдесятью рабочими станциями и несколькими удаленными филиалами на открытое программное обеспечение? Об этом расскажет **Дмитрий Макаров**.

Первый раз я столкнулся с GNU/Linux в 2003 году, это был Red Hat. Особого впечатления на меня он тогда не произвел – видимо, сказалось общая неграмотность в данном вопросе. Однако это было только начало знакомства с захватывающим миром Open Source. Далее интерес усиливался, и открытое программное обеспечение все больше входило в мою повседневную жизнь. Я прошел все этапы на этом пути – вплоть до торжественного форматирования винчестера с Windows и полной замены ОС от Microsoft на GNU/Linux.

Что бы ни говорили некоторые, но в последнее время четко обозначилась тенденция увеличения интереса к решениям на базе открытого ПО. Для большинства офисных потребностей Open Source давно предлагает отличный функционал. А с недавнего времени, с появлением такого продукта как *WINE@Etersoft*, возможностей для перехода на свободные технологии стало намного больше. GNU/Linux и программное обеспечение на его основе давно завоевали лидирующие позиции в серверном сегменте рынка и, развивая свой успех, все больше вторгаются в настольный сектор. Думаю, что Linux уже готов

для использования в качестве основной системы как дома, так и на работе. Но лучше всего GNU/Linux, в силу своей простоты и надежности, подходит для корпоративного применения.

Сегодня компания «АйТи Парус» (**www.itparus.ru**) сосредоточила свои усилия вокруг дистрибутива Gentoo Linux. В нашей работе оказались очень полезными его следующие свойства:

- Gentoo содержит большое количество пакетов, как официальных (стабильных), так и неофициальных (экспериментальных).
- Gentoo очень гибок в адаптации к существующим платформам. Можно легко собрать все нужные пакеты для конкретной архитектуры.
- Gentoo гибок в оптимизации для конкретных задач и позволяет поставить только то, что требуется, и ничего лишнего.
- Gentoo не присуще понятие «проклятие зависимостей», от которого порой страдают RPM-дистрибутивы. Многие пользователи, работающие, скажем, с Red Hat, сталкиваются с тем, что для установки какого-нибудь стороннего пакета нужно удовлетворить зависимости, которые тянут за собой собственные зависимости... и так далее.
- Gentoo позволяет легко создавать встраиваемые инсталляции. Например, терминал оплаты, который можно загрузить с «флэшки» объемом в 64 МБ.
- Gentoo прозрачен в использовании и поддержке – разработчики постарались выдержать все в стиле «UNIX-way».
- Gentoo, по своей сути, является «мета-дистрибутивом». Можно организовать свою собственную уникальную сборку пакетов и осуществлять поддержку только для них.

«Чтобы понять сердце Gentoo», пишет создатель дистрибутива Дэниел Роббинс [Daniel Robbins], «представьте себе пользователя, сидящего перед Linux-системой. Что он хочет делать? Философия Gentoo заключается в том, чтобы разрешить пользователю делать то, что он хочет, без того, чтобы стоять у него на пути». Следует отметить, что некоторые сотрудники нашей фирмы помимо Gentoo используют и Ubuntu (например, он стоит на моем ноутбуке), но это, как говорится, дело вкуса.

Компания «Мидитекс Инвест» (**www.miditex.ru**), являющаяся крупным поставщиком текстильной продукции на юге России (более 14 лет на рынке), обратилась к нам с вопросом о переводе своих машин на открытое ПО. В состав компьютерного парка компании «Мидитекс Инвест» входит более 50 рабочих станций и 9 серверов. Клиентские места расположены в основном офисе в Ростове-на-Дону, четырех филиалах по Южному федеральному округу и одной дополнительной торгово-складской площадке в Ростове.

Проекту предшествовали консультации заказчика с нашими инженерами, в ходе которых было составлено техническое задание и установлены цели, которые предстояло достигнуть. Приведу в краткой форме ТЗ, которое послужило отправной точкой для дальнейшего обсуждения и принятия решений.

- Выбор модели работы с удаленными подразделениями. В результате принято решение о терминальном режиме.
- Выработка идеологии построения информационной сети Заказчика. Результатом стало принятие решения о построении VPN-сети между структурными подразделениями.
- Выбор технологии и ОС для реализации данного проекта. Наш выбор пал на GNU/Linux и открытое программное обеспечение.
- Подсчет стоимости владения готовым решением.
- Обоснование предложений и защита их перед Заказчиком.

Руководство заказчика взвешивало все «за» и «против». Рассматривались разные варианты – в том числе и реализация проекта по переходу на платное лицензионное программное обеспечение. В итоге были достигнуты договоренности о внедрении GNU/Linux.

Основной причиной перехода на открытое ПО стало желание соблюсти лицензионную чистоту, высокая стоимость проприетарных программ и нежелание в будущем доплачивать за организацию дополнительных рабочих мест. Так, стоимость перехода на решения Microsoft была оценена в сумму более 70000 долларов США, не считая затрат на дополнительные рабочие места. Мы выразили мнение, что совокупная стоимость владения ПО от Microsoft будет значительно выше нашей оценки, и руководство компании с этим согласилось.



› Наталья Соколовская, менеджер отдела продаж, работает с документами в OOo.

Также немаловажную роль в принятии решения сыграли доводы высокой надежности и открытости решений на базе свободного ПО. В итоге заказчику удалось сэкономить более 60000 долларов США. Заключение договора на техническую поддержку стало логическим завершением проекта. В результате клиент получил полный спектр услуг: от консультаций до внедрения и поддержки.

Основное число рабочих станций компании сосредоточено в Ростове. Не переводить на открытое программное обеспечение было решено только три компьютера финансовой службы, из-за большого количества специализированных программ. Во всех остальных местах внедрение GNU/Linux было проведено успешно.

В процессе перехода было решено множество промежуточных задач. Особое внимание хотелось бы остановить на использовании *1С* под *WINE@Etersoft*, так как достаточно большое число людей интересуется именно этим вопросом.

На сегодняшний день, *WINE@Etersoft* – это единственное решение, позволяющее организовать надежное функционирование решений *1С* в GNU/Linux. За год эксплуатации данного продукта мы ни разу не пожалели о том, что выбрали его, да и от заказчика поступали только пожелания, но не было никаких жалоб и нареканий. Думаю,



› На экране – 1С под управлением терминального сервера на базе Linux.



› **Сотрудник отдела продаж – главного генератора прибыли в компании. Здесь требуются надежные, проверенные продукты.**

что год – это вполне достаточный срок, чтобы заявить в полный голос: компания Etersoft (**www.etersoft.ru**) предлагает продукт, который действительно позволяет использовать *1C* в среде GNU/Linux. Основной причиной применения его в данном проекте был тот факт, что остальные решения не позволяли полностью уйти от использования ОС Windows.

Переход был спланирован на майские праздники 2007 года. Персонал был уведомлен о грядущих изменениях, и, вернувшись после отдыха, сразу же приступил к работе в новой системе. Реализация была проведена в три этапа:

» **Подготовительный**. На данном этапе рабочие станции были подготовлены к превращению в тонкие клиенты. Там, где это требовалось, сетевые карты были заменены на адаптеры с boot ROM. Был подготовлен сервер для загрузки бездисковых станций и терминальный сервер на базе ОС Gentoo 2007. Также было приобретено и опробовано ПО *WINE@Etersoft SQL Edition*, которое и по сей день используется для запуска продуктов 1С. Была проведена проверка совместимости всего оборудования, используемого Заказчиком. Этап прошел успешно и был закончен в срок. Заказчик убедился в работоспособности решения и дал окончательное согласие.

» **Основной**. Как я уже отметил ранее, данный этап был осуществлен в выходные и праздничные дни совместными усилиями нашей компании и ИТ-отдела заказчика. Все требовалось осуществить в кратчайшие сроки, поэтому помощь оказалась весьма кстати. Предстояло настроить профили всех сотрудников и проверить работу всех программ и доступность принтеров. Этот этап занял два дня. Его завершением была проверка функционирования всех рабочих мест.

» **Авторский надзор**. В ближайший за праздниками рабочий день заказчик приступил к работе. Первый день был жарким, т.к. у пользователей возникали некоторые вопросы, связанные с отличием интерфейсов программ и ОС. Тем не менее, компания отработала без сбоев, и все прошло гладко. В течении последующих дней авторского надзора были решены все возникшие вопросы и проведено «обучение» сотрудников, у которых вызвала затруднения новая среда.

Не все получилось сразу. Например, на подготовительном этапе рассматривался вариант постепенного перевода рабочих станций и как итог – совместная работа GNU/Linux- и Windows-клиентов *1C* с одной SQL-базой. Здесь возникли трудности с блокировками в *1C*. Именно наличие данной проблемы послужило причиной принятия решения о переводе всех рабочих станций на GNU/Linux разом. Стоит отметить, что уже в следующей версии продукта компания Etersoft решила эту проблему.

Заказчик использует в своей работе SQL-версию *1C: Торговля и склад 7.7*. В качестве сервера БД применяется связка *SBS 2003* + *MS SQL 2000*. Предвосхищая вопросы, скажу, что на момент внедрения компания Etersoft еще не выпустила продукт *SELTA@Etersoft*. [универсальный транслятор SQL-запросов из диалекта T-SQL в pgSQL, позволяет использовать PostgreSQL в приложениях, созданных для MS SQL, – *прим.ред.*]

Что касается одной из стандартных проблем перехода, а именно, печати, то стоит отметить, что для всего парка принтеров компании пришлось приобрести принт-сервера. В нашем случае это была продукция компании D-Link. Данное требование возникло вследствие использования терминального сервера и тонких клиентов для рабочих мест сотрудников. В результате, все принтеры компании стали доступными для сотрудников любых структурных подразделений. Отдельной проблемой была также печать этикеток со штрих-кодом: заказчик активно использует маркировку готовой продукции с применением технологии ШК. Пришлось приложить немало усилий, чтобы заставить работать принтеры Zebra 2844 и Zebra 2824, но в результате была получена рабочая связка CUPS и Zebra. Эти принтеры также подключаются через принт-сервер, что позволяет распечатывать этикетки с любого рабочего места в пределах VPN-сети компании.

В результате всех действий компьютерная структура приняла следующий вид:

» VPN для организации доступа удаленных подразделений к терминальному серверу;

» Терминальный сервер на базе GNU/Linux, связка *LTSP+WINE@Etersoft* для запуска *1C*;

» Сервер для загрузки тонких клиентов и организации рабочих столов для них;

» Сервер печати CUPS;

» Принтеры, оборудованные принт-серверами с Ethernet-подключением.

После реализации проекта заказчик получил мощную, масштабируемую и надежную систему на базе открытого ПО.

Следует признать, что в процессе перехода для пользователей изменилось фактически все: внешний вид рабочего стола, офисный пакет, браузер, почтовый клиент и т.п. Однако основным рабочим инструментом для всех сотрудников заказчика была *1C*, а она никак не изменилась, и это внесло фактор стабильности.

Безусловно, из-за смены привычного окружения были и «недовольные». В этом вопросе огромную роль сыграла волевая позиция руководства: всем дали понять, что пути обратно не будет. Фирма функционировала в прежнем режиме и приносила прибыль, а «консервативность» некоторых сотрудников в конце концов победили доброта и терпение. LXF



LINUX FORMAT
Учебники

Наши эксперты помогут вам с любым приложением Linux



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Большой брат уже следит за нами?

Гаури: Эй, Лина! Можно спросить?
Лина: Что?
Гаури: Кто из них плохие парни?

Из аниме Slayers (Рубаки).

В 2003 году транспортная компания TfL (Transport for London) представила на суд общественности бесконтактный электронный билет Oyster card, использующий технологию RFID. Очень удобная система с защитой от кражи, возможностью пополнения по мере необходимости и прочим. К марту 2007 года уже более 80 % от всех поездок в лондонском метро и автобусах оплачивалось с помощью Oyster. В том же 2007 году обслуживающая Oyster инфраструктура начала целенаправленно переводиться на открытое ПО и протоколы. Это позволило TfL сократить размер лицензионных выплат более чем в пять раз, в частности, из-за ликвидации искусственной привязки к закрытым решениям определенных компаний. (*См. также Новости на стр. 4*)

Все замечательно, все довольны. Но почему Ричард Мэттью Столлмен опять шумит? ZDNet.co.uk опубликовал выдержки из его письма в заметке «Stallman attacks Oyster's 'unethical' use of Linux». Столлмен не изменился – он просто указывает на то, что технология RFID создает инфраструктуру для потенциального нарушения свободы, и ему неприятно использование FOSS в этом процессе. В каком месте нарушение? Дело в том, что билеты не анонимны и информацию о передвижении (данные по поездкам за последние 8 недель) с них может читать кто угодно, в том числе, и без ведома хозяина. Очевидное нарушение неприкосновенности частной жизни, хоть и ради удобства.

При некоторых ухищрениях пассивные метки RFID можно считывать на расстоянии свыше 200 метров, и их использование в виде хранилища личной информации будет только расширяться. Вы готовы предоставлять ее абсолютно всем желающим?

E.M.Baldin@inp.nsk.su

В этом выпуске...

Совет месяца: MySQL

Администратор-root *MySQL* и суперпользователь root используют разные пароли, и если вы забыли первый, вам может показаться, что все пропало. Но если у вас есть стандартный root-доступ к машине, на которой запущен сервер *MySQL*, сбросить пароль вовсе не проблема.

Остановите службу *MySQL* командой

```
/etc/init.d/mysql stop
```

и запустите демона снова, вот так:

```
mysqld_safe --skip-grant-tables &
```

Вы прикажете серверу не проверять пароли – это определенно небезопасно, так что поторопитесь! Войдите как root и установите себе новый пароль:

```
mysql -u root
use mysql;
update user set
password=PASSWORD("НОВЫЙ-ПАРОЛЬ-ROOT") where
User='root';
flush privileges;
quit
```

Остановите сервер и вновь запустите его командами:

```
/etc/init.d/mysql stop
/etc/init.d/mysql start
```

а затем используйте новый пароль root для общения с СУБД обычным образом. Вы изменили пароль только у root – остальных пользователей это не касается, и приложения, базирующиеся на *MySQL*, такие как блоги, CMS-системы и *MythTV*, ничего не заметят.

KDE: Приберем

Энди Ченнел наносит финальные штрихи на свой персональный рабочий стол.





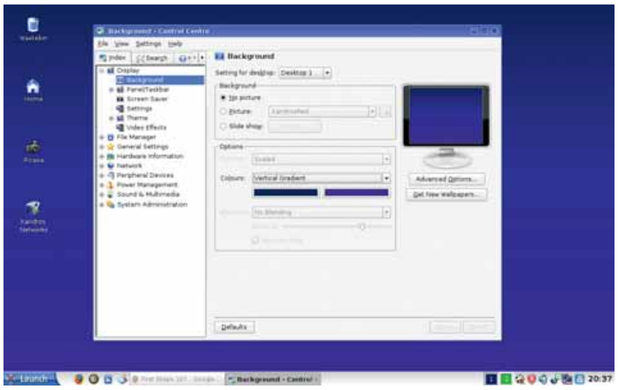
Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется со времен Dragon 32.

Машины имеют репутацию холодных и безличных, но существуют тысячи вариантов придания окружениям KDE и Gnome некоторой индивидуальности. Первое, что вы видите после загрузки – это рабочий стол, и, в зависимости от выбранного вами дистрибутива, тут может быть абстрактный рисунок, изображение дерева или нечто броское. Стандартные фоны разрабатываются не для того, чтобы посягнуть на ваше творчество во время работы, а потому обычно непритязательны и – да! – скучны. Однако любая настольная ОС имеет опцию для замены этого изображения чем-то более личным – и это прекрасная возможность напомнить себе о семье, если вы погрязли в сидении за экраном.



› Рабочие столы компьютеров – но не с Ubuntu – обычно окрашены в мягкие синие тона с разводами.



› Градиентный рабочий стол – лучший компромисс между простотой и транжирством ресурсов для маломощных машин.

Чтобы изменить картинку в KDE, необходимо посетить Центр управления: он или спрятан в меню Система, или занимает особое место в главной части K-панели, появляющейся при нажатии кнопки K-меню. После открытия Центра управления выберите в левой панели пункт Внешний вид и темы, а затем Фон. Опций здесь немало, но вполне очевидно, что следует щелкнуть на выпадающем списке рядом с надписью Изображение и выбрать новые обои. Список просто показывает все изображения из **/usr/share/wallpapers/**, и мы легко можем поместить в этот каталог понравившиеся картинки, а потом выбрать одну из них. Нажатие кнопки Применить внизу диалогового окна обновит фоновое изображение.

Картинки на рабочем столе

Однако мы не обязаны ограничиваться изображениями только из одного каталога. Для выбора снимков из других мест, щелкните на иконке каталога рядом с ранее использованным списком выбора и перейдите в место расположения вашего любимого изображения. Выберите его, нажмите Применить, и фон тут же обновится.

Пониже пункта выбора изображений имеется несколько опций, используемых для центрирования, растягивания или размещения изображения мозаикой по всему экрану. Для маломощных машин, вероятно, будет лучше вообще удалить фон. Если мы это сделаем, то начнут действовать опции цвета в срединной секции и станет возможным создать градиент, задав два цвета и выбрав Вертикальный градиент в выпадающем списке Цвета. Эта опция дает слегка отдохнуть уставшим глазам без чрезмерной нагрузки на системные ресурсы.

В верхней части данного диалога находится опция для настройки различных фонов для каждого виртуального рабочего стола, а справа – кнопка Дополнительно, позволяющая изменить цвет текста ярлыков рабочего стола или даже добавить тень, чтобы сделать белый текст более читаемым на светлом фоне. Фон рабочего стола – всего лишь одна опция; KDE способен и полностью изменить свой внешний вид, включая заголовки окон, полосы прокрутки и виджеты приложений – кнопки, флажки и выпадающие списки, образующие интерфейс поль-

Месяц назад Мы от души порезвились в замечательных Картах Google и *Google Планета Земля*.

рабочий стол

зователя. Все вместе эти косметические подстройки группируются в «темы», доступные в Менеджере тем Центра управления.

Давайте начнем наше изучение тем с Декораций окон – таких вещей, как кнопки минимизации и заголовок. Пользователям, перешедшим с Windows, стандартное расположение элементов строки-заголовка окна (название и иконка приложения – слева, кнопки свернуть, развернуть и закрыть – справа) сразу покажется знакомым. Но если вы привыкли к OS X на Mac, то вам, вероятно, ради своего удобства захочется внести изменения – к счастью, это легко.

В стиле Mac

В разделе Декорации окон выберите По центру для Выравнивания заголовка и нажмите кнопку Применить [Наличие данного пункта зависит от выбранной темы оформления, – прим. пер.]. Название каждого окна теперь будет располагаться по центру строки-заголовка. Для работы с кнопками, перейдите на вкладку Кнопки и выберите Использовать свое расположение кнопок заголовка окна. Выбор данной опции активирует шаблон строки заголовка в центре экрана, позволяя переносить кнопки к левому или правому краю строки. Для создания Mac-атмосферы мы должны передвинуть кнопки Закрыть, Свернуть и Развернуть к левому краю и добавить кнопку Меню из списка под шаблоном к правому краю. Любые выполненные изменения будут автоматически отображаться на примере строки заголовка внизу данного окна. Завершив редактирование, нажмите кнопку Применить, чтобы изменения вступили в силу.

Оформив окна, можно изменить раскраску, выбрав опцию Цвета в левой панели и настроив оттенки на свой вкус. И вновь, имеется несколько уже готовых настроек – и мы можем изменить цвет отдельных элементов, выделив их имена в выпадающем меню раздела Цвет виджетов и выбрав соответствующий цвет. Завершив макияж, нажмите кнопку Сохранить схему, задайте имя и нажмите Сохранить.

Приступим к созданию стандартных ярлыков-иконок для часто вызываемых приложений. Для этого сначала следует определить, где размещаются программы, которые надо запускать. Есть два способа сделать это: найти приложение в файловой системе – скорее всего оно расположено где-то в **/usr/bin** – или получить информацию через редактор меню KDE. Для этого вновь откройте Центр управления, затем выберите Панели, щелкните на вкладке Меню и затем нажмите Редактировать меню K.

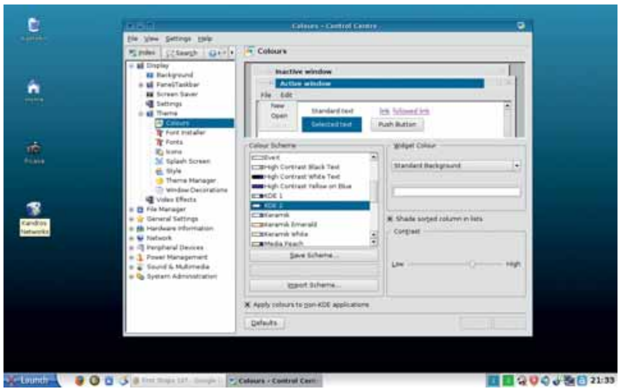
Теперь можно порыскать по структуре меню и выбрать приложение, чтобы просмотреть необходимую нам команду его запуска. Для *Firefox*, например, это **firefox %u**, которая и потребуется для создания ярлыка. Зная ее, мы теперь можем щелкнуть на рабочем столе правой кнопкой мыши и выбрать Создать > Ссылка на приложение. В появившемся диалоге прежде всего следует ввести имя, а затем щелкнуть на кнопке Пиктограмма для выбора иконки. Если ни одна стандартная иконка не подходит, можно использовать любой PNG-файл – просто выберите Другие и нажмите кнопку Обзор, чтобы найти подходящий. Вкладку Права доступа можно проигнорировать и перейти сразу же к Приложение. Здесь мы вводим описание и комментарий (появляющийся в подсказке), а затем добавляем команду в соответствующую строку. Затем нажимаем OK, и ярлык будет добавлен на рабочий стол, вместе с именем и подсказкой. Более того, его можно перенести на панель внизу экрана, чтобы ярлык был доступен всегда. [Ярлык переносится на панель быстрого запуска, а уж где она будет расположена, решает пользователь. Кроме того, имеется более простой способ: нажимаем кнопку K-меню, затем находим ярлык приложения, щелкаем по нему правой кнопкой мыши и выбираем Добавить элемент на рабочий стол, – *прим. пер.*]

Web-ярлыки

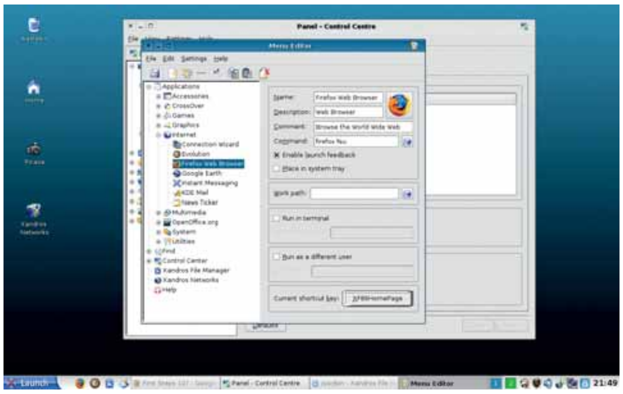
Давайте сделаем еще один шаг: заставим *Firefox* запускаться с какой-либо конкретной web-страницы. Нам следует лишь открыть свойства ярлыка (правый щелчок и Свойства) и добавить URL к команде на вкладке Выполнить: **Firefox** http://www.facebook.com. Однако, если *Firefox* уже запущен, это приведет к открытию сайта в новой вкладке, а не в собственном окне, что сводит на нет наличие отдельного ярлыка. Так что нам необходимо указать в строке запуска аргументы команды, гарантирующие, что она откроет новое окно. Добавьте следующее: **Firefox -new-window** http://www.facebook.com и нажмите OK. При двойном щелчке на таком ярлыке web-сайт или сервис откроется в новеньком окошке. LXF

Скорая помощь

Не впадайте в крайность со сложными текстурами на рабочем столе – они лишь затруднят поиск файлов.



› **Кто сказал, что синий цвет – это обязательно? Раздел Цвет дает нам целую радугу на выбор.**



› **Добавить ярлык для вашего любимого приложения на рабочий стол легко. Команда для запуска *Firefox* спрятана где-то здесь.**

Konqueror: Его

Пусть комбинированный файл/web-браузер KDE пляшет под вашу дудку.

Скорая помощь

Вы можете настроить более продвинутые действия в web посредством настроек JavaScript, что может быть использовано для подавления всплывающих окон и других неприятностей.

Вы спросите: а зачем нужно мучиться с *Konqueror*? Ведь это комбинированный файл/web-браузер, используемый старой версией KDE, а последняя версия KDE перешла на новый файловый менеджер *Dolphin* и неоспоримый король web – *Firefox*. Что ж, ответ такой: *Konqueror* – чрезвычайно мощное приложение, которое поможет вам раскрыть потенциал вашей машины.

На данном уроке мы собираемся рассмотреть скрытые глубины *Konqueror* (KDE версии 3.5) и показать, как можно в несколько щелчков переключиться от web-представления к локальному; убедиться, что *Konqueror* открыт в нужном режиме; и как настроить отдельные версии режимов браузера для каждого конкретного пользователя или применить модели. Это превосходно, если, к примеру, у вас есть отдельные наборы любимых вкладок для работы и для досуга или если вы хотите создать для каждого члена семьи собственный профиль браузера, не связываясь с полностью многопользовательской системой. Преимущества этого подхода в том, что профиль каждого пользователя может иметь свою собственную домашнюю страницу(ы), историю, cookie и закладки, и вы, возможно, сэкономите системные ресурсы и время, поскольку переключение профиля занимает намного меньше времени, чем переключение пользователя.

Выбрав *Konqueror* в качестве web-браузера, вам первым делом следует настроить домашнюю страницу. Перейдите на сайт, который хотите использовать в данном качестве, выберите Настройки > Сохранить профиль просмотра «Простой браузер» и нажмите OK. В следующий раз при запуске браузера он отобразит новую домашнюю страницу. После этого можно даже заставить приложение запомнить размер окна, установив соответствующую галочку перед тем, как нажать кнопку Сохранить.

Настройка Web

Разработчики старались сделать *Konqueror* как можно более гибким, что выразилось в весьма устрашающем виде диалога опций, появляющегося, когда вы выбираете Настройки > Настроить Konqueror; окно переполнено информацией. Но нам интересны только несколько опций настройки для web-браузера, поэтому щелкнем на опции Поведение Web слева. Она намного проще, и мы рассмотрим все сверху донизу.



› Настройка в *Konqueror* просмотра web может серьезно озадачить – лучше заняться только простыми вещами!

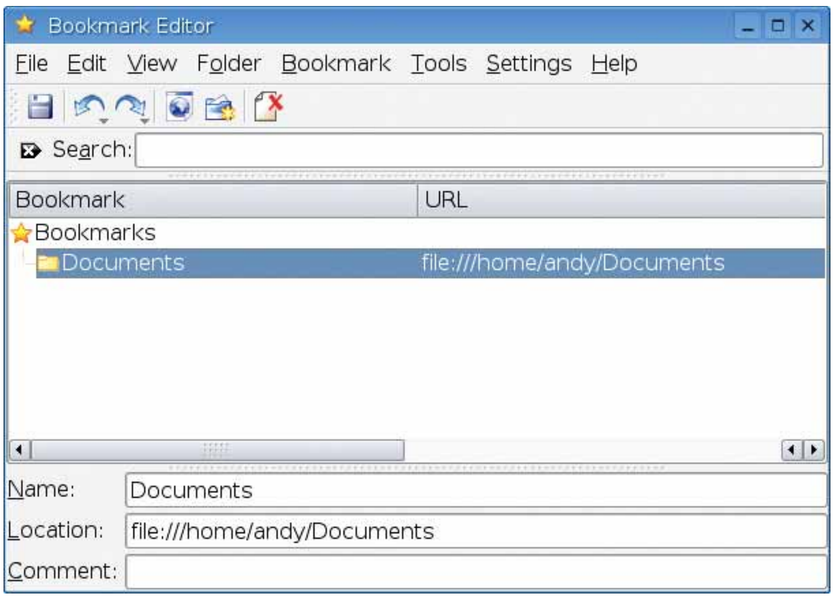
Первая опция, Закладки, понравится людям с организаторским зудом, поскольку она дает возможность самому указывать каталог и имя каждый раз при создании закладки, а не перекладывать это на сайт или web-страницу. Вторая опция определяет, как *Konqueror* отображает закладки в Панели закладок, которую вы можете вызвать, выбрав Настройки > Панели инструментов > Панель закладок (Konqueror). По умолчанию все подряд сохраняется в каталоге закладок, но выбор этой опции позволяет вам настроить видимость на уровне страниц. Чтобы пометить закладку для отображения на панели инструментов, выберите Закладки > Изменить закладки, щелкните на соответствующей записи и выполните Правка > Показывать на панели инструментов. Также стоит перейти в меню Настройки редактора закладок и выбрать Автосохранение при выходе, или вам придется вручную сохранять изменения перед закрытием редактора. Опция Заполнение форм просто определяет, будет ли *Konqueror* запоминать такие детали, как электронные адреса, имена пользователей, телефонные номера и т. д., а затем добавлять их в web-формы.

Экономим место

В разделе Навигация с закладками выберите первую опцию, чтобы *Konqueror* открывал новые окна во вкладках – полезная экономия места – а если вам необходим больший контроль, откройте окно Дополнительно.

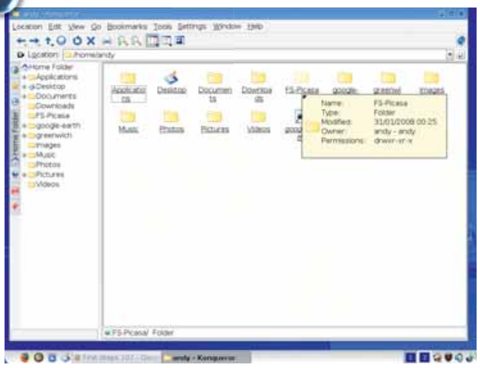
Опции нижней половины окна менее важны, но описывают такие элементы, как изменение указателя мыши при наведении на ссылку, автоматическая загрузка изображений (выключите ее, если у вас медленное соединение) и изменение настроек подчеркивания ссылок. Последний раздел, Анимация, позволит вам заставить назойливые анимированные GIF'ы проигрываться лишь раз или не проигрываться вообще.

› В режиме Менеджера файлов, *Konqueror* все еще может создавать закладки для быстрого и легкого доступа к конкретным местам системы.



легко настроить!

Шаг за шагом: Настроим Konqueror как web-браузер



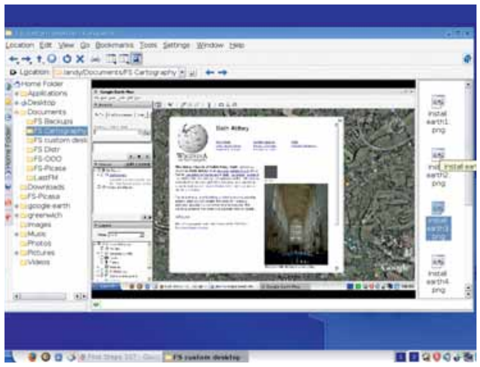
1 Запуск Konqueror

Дистрибутивы, использующие *Firefox* как стандартный браузер, скорее всего, спрятали *Konqueror* в меню Утилиты или Аксессуары, и при запуске он, видимо, предстанет в своем файловом воплощении. Если вы не можете найти его в меню, выберите опцию Выполнить в K-меню (или откройте терминал) и введите 'konqueror'. Многие другие приложения можно запустить подобным образом.



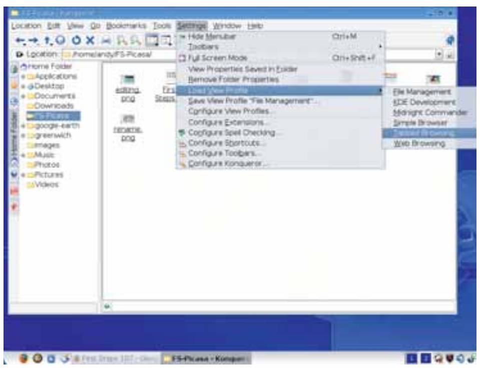
2 Предпросмотр изображений

В режиме файл-браузера можно превратить каталог, полный изображений, в мгновенное слайд-шоу, щелкнув дважды на первом изображении и затем выполнив Инструменты > Слайд-шоу (или нажав F2). После этого каждое изображение будет показываться несколько секунд перед переключением к следующему. Если вы предпочитаете ручное управление при просмотре, используйте для перемещения вперед пробел, а назад – Backspace.



3 Визуальная навигация

Konqueror также имеет специальный режим Фотоальбом, его можно использовать для визуальной навигации по коллекции изображений. Он включается при помощи меню Вид > Режим просмотра > Фотоальбом. При такой настройке, главное окно будет отображать большие изображения. Небольшая панель справа отображает миниатюры содержимого каталога, а навигацию можно осуществлять при помощи древа файлов слева.



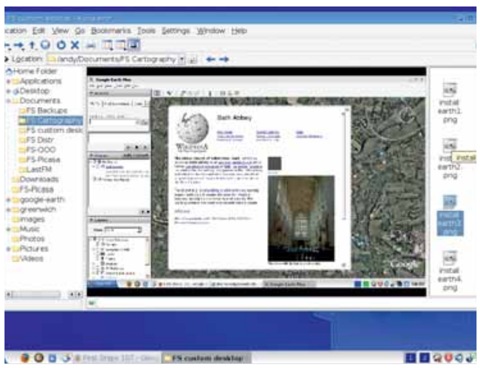
4 Выбор режима браузера

Разработчики *Konqueror* создали несколько профилей, подходящих для web (простой и с закладками), файлов и FTP. Получить доступ к ним можно через Настройки > Загрузить профиль просмотра. Мы собираемся подработать профиль Навигация с закладками, так что выберите его и посмотрите, как файл-менеджер превратится в web-браузер с главными KDE-сайтами на вкладках.



5 Выберите свою страницу

Пройдитесь по всем вкладкам по очереди и измените web-адрес на тот, что вы хотите использовать. Можете добавить новую вкладку, щелкнув на пустом месте после последней вкладки. Теперь выполните Настройки > Сохранить профиль просмотра «Навигация с закладками». Убедитесь, что вы выбрали Сохранить URL в Профиле в появившемся диалоге и нажмите Сохранить. В следующий раз при выборе этого профиля ваши сайты откроются автоматически.



6 Многопользовательский

На одном компьютере можно иметь несколько различных профилей. Используя те же действия, что и выше, заполните профиль Навигация с закладками, а затем, в конце, введите имя профиля в строке Имя профиля и нажмите Сохранить. Новый профиль добавится к вашему списку, и его можно будет выбрать обычным способом. LXF

» **Через месяц** Ожидание было долгим, но Firefox 3 наконец-то здесь. За дело!

Eee PC: Добавим

Часть 1: Майк Сондерс объяснит, как превратить ваш Eee PC в идеальную рабочую мини-станцию, обогатив установку по умолчанию новыми программами.



Маленький, легкий, дешевый и под Linux – ПК Asus Eee за $450 (http://eeepc.asus.com) породил новую тенденцию в мире компьютеров, заставив конкурентов наперегонки выпускать дешевые микроноутбуки. В теме номера прошлого месяца мы спрогнозировали влияние Eee на рынок: он познакомит миллионы новых пользователей с Linux, и они обнаружат, что Windows не есть все и вся в компьютерном мире. Мы также рассказали, как приодеть ваш рабочий стол меню программ и различными фоновыми рисунками, а также как освободить место на панели задач.

На сей раз мы запускаем новую серию из трех частей, посвященную Eee. Если вы счастливый обладатель исходной модели 701 или новой версии 900, вы обнаружите, что там есть много чего для исследования. Хотя набор приложений, установленных по умолчанию – *OpenOffice.org*, *Firefox*, *Pidgin* и т.д. – порадует любого рядового пользователя, полный дистрибутив Linux, Xandros, вопиет о большем. На данном уроке мы покажем, как добавить приложений в ваш Eee и поменять рабочий стол *IceWM*, установленный по умолчанию, на полную установку KDE. Так что кладите ПК размером с пивную кружку на колено, подключайтесь к ближайшей точке Wi-Fi, и начнем...



Наш эксперт

Майк Сондерс урвал Eee PC еще в ноябре, и перекроил приложения всеми возможными способами. Он носится со своим ноутбуком как с маленьким пластмассовым ребеночком.

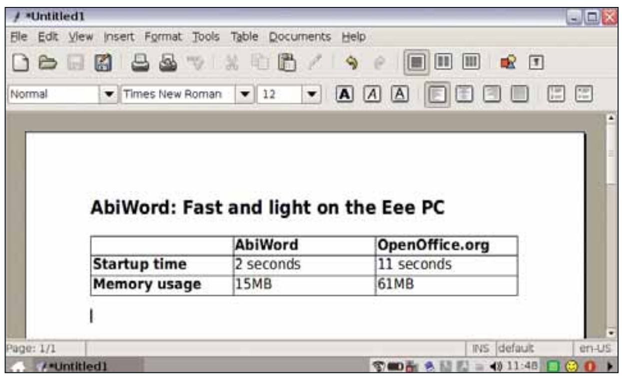
Часть 1 Установка дополнительных приложений

Как уже упоминалось, Eee управляется Xandros Linux, дистрибутивом, построенным на солидной основе Debian, но по популярности и близко не лежащий к Ubuntu со товарищи. Asus подписал контракт с Xandros на разработку специальной версии дистрибутива для Eee – кое-что было удалено, было сделано множество специальных настроек, но в целом это все-таки Xandros Linux. Если вам хорошо знакомы пакеты и репозитории, вы можете пропустить несколько следующих параграфов и перейти к добавлению нового репозитория. Но если вы новый пользователь, взявший наш журнал в руки благодаря Eee, внимайте!

Приложения Linux поставляются в пакетах. В Windows для установки программы вы обычно дважды щелкаете по файлу **setup.exe**, а в Linux вам нужно скачать тщательно собранный архив, содержащий основные части программы – исполняемый код, изображения, документацию и прочее. В случае Xandros Linux эти файлы пакетов имеют расширение **.deb**, так что, например, web-браузер *Firefox 2.0* поставляется как **firefox_2.0.deb**.

Короче, Deb-пакеты являются неким подобием exe-файлов Windows, разве что вы не можете выполнять их напрямую – а с точки зрения безопасности это очень хорошо. Deb'ы устанавливаются посредством менеджера пакетов, который называется *Synaptic*. Но вот еще что: зачастую работа программы будет зависеть от других приложений. Большая часть Windows-программ включает все необходимое в **setup.exe**, что приводит к раздутости и массовому дублированию кода. В Linux имеется система «зависимостей» – то есть Deb-пакетов, от которых зависят другие Deb-пакеты. Например, пусть на вашей машине есть пять программ, которые могут отображать HTML-документы. В мире Windows каждая из этих программ будет иметь свой собственный движок отображения HTML, а в Linux все они будут зависеть от внешнего пакета, выполняющего эту работу, чем устраняется дупликация кода [пример несколько натянут, но суть схвачена верно, – *прим.ред.*].

Помнить о всех этих зависимостях – тяжкий труд, но Linux облегчает его, используя репозитории. Это онлайновые коллекции приложе-



› **Достала раздутость и медительность *OpenOffice.org*? Добавьте репозитории с новыми пакетами и установите *AbiWord*.**

» **Месяц назад** Как настроить GUI Eee.

приложений



› При размерах 22,5x16,4x3,5 см и весе всего 0,92 кг, Eee идеален для поездок.

Web-камера в Продвинутом режиме

Если вы перейдете в Продвинутый режим, не удивляйтесь пропаже утилиты для web-камеры. Программа никуда не делась – она просто сменила имя. Нажмите кнопку Launch слева внизу, перейдите в Applications > Multimedia > UCView, и сможете делать снимки и записывать фильмы, как и раньше.



› В Продвинутом режиме (см. стр. 61), программа web-камеры называется *UCView*.

ний, где вместе хранятся программы и их зависимости. Итак, мы собираемся добавить новый репозиторий на наш Eee: это позволит скачать дополнительные программы (а также их зависимости), без необходимости просматривать тьму-тьмущую сайтов.

Добавление репозитория

По умолчанию, Eee настроен на скачивание приложений только с одного сайта: официального репозитория обновлений Asus Eee. Он используется для исправлений ошибок и заплаток безопасности, но не содержит никаких программ сверх того, что поставляется с машиной. А мы хотим добавить новый репозиторий с сотнями новых пакетов – приложений, которые вы хотели бы видеть в любом типовом дистрибутиве для настольного ПК. Для этого вам нужно, чтобы ваш Eee был подключен к Интернету.

На рабочем столе нажмите Ctrl+Alt+T, чтобы открылось окно командной строки. Теперь введите следующее:

```
sudo kwrite /etc/apt/sources.list
```

Здесь мы запускаем текстовый редактор *KWrite*, чтобы изменить список репозиториев, которые может использовать наш Eee. (Слово **sudo** означает, что мы хотим редактировать файл как суперпользователь, он же администратор, поскольку обычным пользователям нельзя изменять системные файлы.) Вы увидите, что файл содержит две строки – это расположения Asus'овских обновлений безопасности. Добавьте к файлу третью строку:

```
deb http://xnv4.xandros.com/xs2.0/upkg-srv2 etch main contrib non-free
```

Убедитесь, что строка скопирована правильно, иначе потом вы получите сообщения об ошибках. Это репозиторий Xandros Server'a, дистрибутива Linux, на котором основана версия Eee. Следовательно, он отлично совместим с пакетами для Eee, и ваша система не придет в беспорядок, если вы ограничитесь установкой оттуда обычных программ (но не пытайтесь обновить ядро или важные системные скрипты).

Сохраните файл (File > Save) и закройте *KWrite*. Вернитесь обратно в окно терминала и введите:

```
sudo apt-get update
```

Эта команда запрашивает список доступных пакетов из онлайн-репозитория Xandros и, соответственно, обновляет базу данных вашего Eee. (Если вы увидите предупреждающие сообщения о цифровых подписях GPG, можете смело их игнорировать.) Теперь все готово для установки новых приложений!

Существуют сотни доступных пакетов – вот некоторые наиболее интересные:

» **AbiWord** Быстрый и легковесный *MS Word*-совместимый текстовый процессор, прекрасная альтернатива массивному *OpenOffice.org*.

» **Gnumeric** Основанный на *Gtk* редактор электронных таблиц, куда более быстрый, нежели его аналог из *OOo*.

» **Эмуляторы** Eee поддерживает эмуляцию старых (8/16-битных) консолей и компьютеров. См. *Snes9x* (SNES), *Dgen* (Mega Drive/Genesis), *Nestra* (NES) и *Vice* (C64).

» **Средства разработки** Если вам надо накропать немного кода на C/C++, установите пакет **build-essential**. Это мета-пакет, устанавливающий *GCC*, *Binutils*, *GNU Make* и файлы разработки *Glibc*.

»

» **Серверное ПО** Вы можете даже запустить на вашем Eee *Apache*, PHP и *MySQL*, для web-разработки на ходу.

Эти программы, а также многие другие, можно установить из командной строки, с помощью **sudo apt-get install <программа>**; но более изящный способ, также облегчающий их поиск – воспользоваться менеджером пакетов *Synaptic*, применяемым в Xandros, Ubuntu и других системах Linux на базе Debian.

Установив приложения, можете закрыть *Synaptic*. Взгляните на категории по вкладкам вашего рабочего стола: некоторые графические программы, типа *AbiWord* (вкладка Work), автоматически добавляют иконки быстрого запуска. Но если новая программа не создала значок, для запуска вам придется нажать Ctrl+Alt+T, а затем ввести ее имя.

Создание новых значков на рабочем столе вручную – сущая каторга; куда проще добавить на панель задач меню Start и изменять его. Мы описали этот процесс в прошлом номере, но если у вас под рукой его нет, вот краткое содержание:

1 Нажмите Ctrl+Alt+T и введите **sudo KWrite /etc/X11/IceWM/references**.

2 Нажмите Ctrl+F и найдите TaskBarShowStartMenu (регистр важен) в файле **prefs**.

3 Поменяйте строку TaskBarShowStartMenu=0, чтобы она заканчивалась на =1.

4 Нажмите Ctrl+S, чтобы сохранить файл, закройте редактор и перезапустите Eee.

Теперь вы видите меню Start в левом нижнем углу экрана. Это меню *IceWM* по умолчанию; оно заполнено программами, которых у вас, возможно, и нет, но его легко редактировать. В окне командной строки введите **sudo kwrite /etc/X11/Icewm/menu** и измените этот файл по своему вкусу. Формат строк меню таков: «Имя в меню, значок, команда». Если у вас нет подходящего значка, используйте 'app'. Например, так добавляется в меню *AbiWord*:

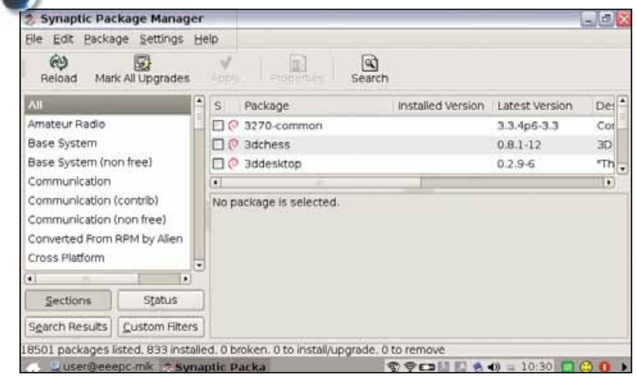
```
prog AbiWord app abiword
```

Чтобы удалить программу через *Synaptic*, найдите ее при помощи функции поиска, щелкните по ее имени в списке пакетов правой кнопкой мыши и выберите 'Mark for Complete Removal'. После нажатия Apply, программа и файлы ее конфигурации будут удалены с Eee.

Скорая помощь

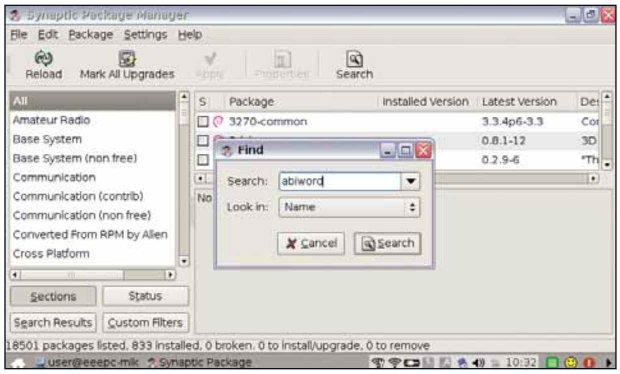
Если у вас появилось окно, размер которого больше экрана, и вы хотите получить доступ к кнопкам в его нижней части, нажмите и удерживайте левую клавишу Alt, затем щелкните в окне и оттащите его мышью. Это работает как в Простом, так и в Продвинутом режиме.

Шаг за шагом: Установка приложений через Synaptic



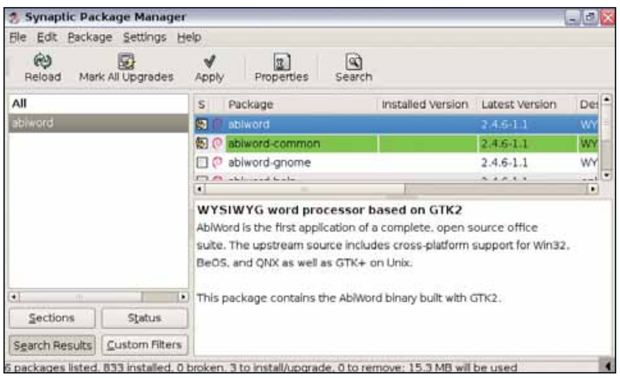
1 Запуск Synaptic

Нажмите Ctrl+Alt+T, чтобы открылся терминал, затем введите **sudo synaptic**. Вы увидите это окно; щелкните по списку категорий слева, чтобы осмотреть доступные приложения.



2 Поиск приложений

Чтобы найти конкретную программу, щелкните по кнопке Search вверху. Введите имя искомой программы, а затем нажмите Search. Через пару секунд вы увидите список программ.



3 Установка нового приложения

Для установки программы, щелкните по флажку рядом с ее именем и выберите 'Mark for Installation'. (Не волнуйтесь, если появится сообщение о том, что пакеты не идентифицированы). После этого нажмите Apply.



4 Подтвердите изменения

Вы увидите этот экран подтверждения: используйте Alt+левый щелчок, чтобы переместить окно повыше и сделать видной кнопку Apply, после этого нажмите ее. Новая программа/ы будет установлена из Интернета.

Часть 2 Продвинутый рабочий стол

Хотя *IceWM*, менеджер окон Eee по умолчанию, весьма быстр и идеален для многих пользователей, вам, возможно, понадобится рабочий стол Linux побогаче. Eee поставляется с большим кусками KDE, включая библиотеки и утилиты, но вот самого рабочего стола нет. Это не проблема: если ваш Eee подключен к Интернету, вы можете заполучить полную установку KDE, выполнив несколько команд и скачав нужные модули. На языке Eee это называется Продвинутый режим [Advanced mode].

Мы собираемся настроить Eee так, чтобы можно было переключаться между Продвинутым режимом и Простым рабочим столом [Easy Desktop] (*IceWM*). Отметим, что KDE потребляет существенно больше памяти, чем *IceWM*; когда он в полном размахе, то может занимать 30–100 МБ ОЗУ, в зависимости от состава запущенных приложений. Однако тех 512 МБ, что имеются у Eee, хватит для одновременного выполнения KDE, *Firefox* и *OpenOffice.org*, и если у вас нет приложения, съедающего весь ОЗУ до последнего байта, вы не ощутите существенного дефицита производительности от использования KDE.

Включение Продвинутого рабочего стола требует немного больше работы в командной строке, но, как мы уже видели, это не смертельно. Если у вас CLI-фобия – просто следуйте этим пунктам...

1 Нажмите Ctrl+Alt+T, чтобы открыть окно командной строки.

2 Введите следующие команды:

```
wget http://download.tuxfamily.org/eeepcrepos/key.asc
sudo apt-key add key.asc
rm key.asc
```

Вы скачаете и установите ключ безопасности для пакетов, которые мы намерены установить, для уверенности в том, что они законные и не были подделаны.

3 Затем, введите **sudo kwrite /etc/apt/sources.list** и внизу списка добавьте такую строку:

```
deb http://download.tuxfamily.org/eeepcrepos p701 main
```

Сохраните файл и выйдите из редактора. Строка скажет системе управления пакетами, где достать дополнительные приложения, требующиеся для Продвинутого режима.

4 Введите **sudo apt-get update**, чтобы получить список пакетов, которые теперь можно скачивать.

5 И, наконец, введите

```
sudo apt-get install advanced-desktop-eeepc
```

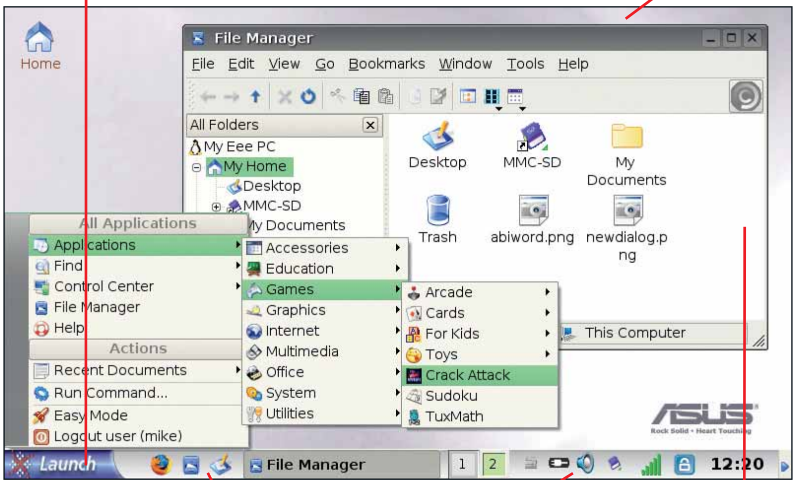


› **Задействовав Продвинутый режим, в диалоге выхода вы увидите новую кнопку Full Desktop.**

Исследуем Продвинутый интерфейс

Меню
Щелкните по Launch, и откроется главное меню для приложений, а также для выхода и выключения.

Рабочий стол
Сюда можно добавлять запуск новых программ и сохранять файлы для быстрого к ним доступа.



Значки
Эти кнопки запуска дают быстрый доступ к вашим любимым приложениям.

Лоток
Системный лоток содержит значки-извещения о внешних устройствах и беспроводном соединении.

Файлы
Как и в Простом режиме, вашими документами управляет файловый менеджер Xandros.

Вас спросят, скачать ли пакеты (введите Y), что займет примерно 10 секунд при DSL-соединении (а при телефонном – несколько минут).

Все готово! Теперь у вас есть все приложения, требующиеся для Продвинутого рабочего стола; давайте попробуем его. Нажмите на кнопку выключения вашего Eee, и в появившемся диалоговом окне вы увидите новый значок под названием Full Desktop – см. снимок экрана слева внизу на этой странице. Щелкните по кнопке – Eee перезапустит X-сервер (графический интерфейс), и еще спустя несколько секунд вы увидите рабочий стол KDE. Если вы регулярный пользователь KDE, у вас не будет проблем с его использованием – он сильно перегружен логотипами Xandros и Asus, но в целом это стандартная установка KDE.

Если KDE вам в новинку, вы сразу поймете, что это более гибкий рабочий стол, чем *IceWM*. В нижней панели задач можно щелкать правой кнопкой мыши по значкам, чтобы смещать или удалять их; и вы, конечно, удалите несколько значков, чтобы полностью увидеть часы! KDE гибко настраивается: с помощью утилиты *KDE Control Centre* вы можете менять шрифты, цвета, оформление окон и темы виджетов, там также отображается информация об аппаратной части вашего Eee.

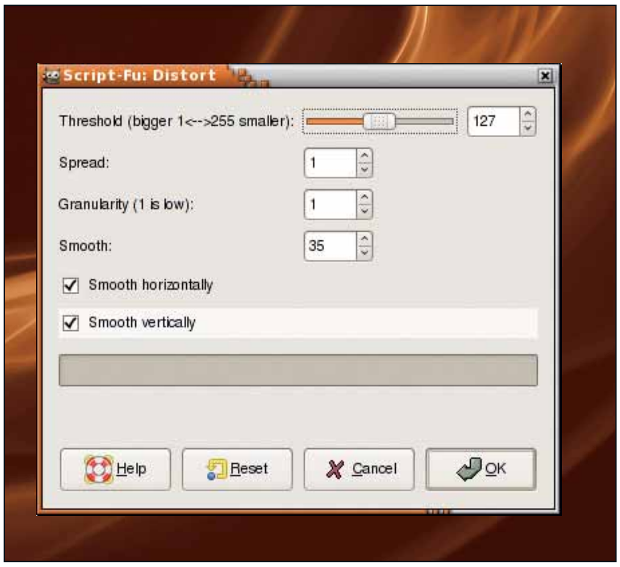
Посмотрите на снимок экрана для беглого обзора интерфейса. Если вы хотите вернуться в Простой режим, щелкните по кнопке Launch (слева внизу), а затем в появившемся меню выберите Простой режим. Многие пользователи Eee любят иметь под рукой оба рабочих стола: Простой режим для быстрого обзора web и заметок, а Продвинутый режим – для украшательств полной установки Linux. Будьте готовы к следующему номеру, когда мы рассмотрим специальный дистрибутив Breeezy Eee, и способы резервирования данных вашей машины. LXF

» **Через месяц** Резервирование вашего Eee и загрузка других дистрибутивов Linux.

Gimp: Добавляем

Пока не изобрели обмен мгновенными сообщениями, люди посылали письма и запечатывали их воском. **Майкл Дж. Хэммел** покажет, как оцифровать вашу восковую печать.





› **Может потребоваться изменить настройки Сглаживания. Заполнив выделенную область цветом, мы к ним вернемся и все исправим.**



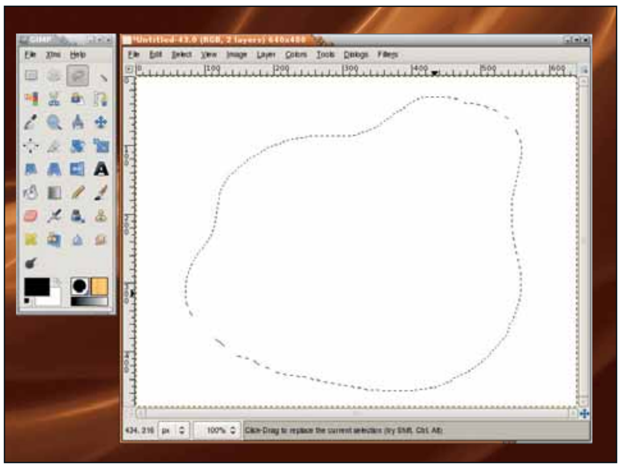
Наш эксперт

Майкл Дж. Хэммел является участником проекта *GIMP* и автором трех книг по данной теме, включая последнюю: *The Artist's Guide to Gimp Effects* [Введение в эффекты *GIMP* для художников].

Возобновив серию уроков по *GIMP*, я хочу подстрекнуть читателей *Linux Format* к выходу за рамки создания в нем банальных настольных картинок. Большую часть эффектов *Photoshop* можно применять и в *GIMP*, и это не удивительно: по сути, оба приложения работают с пикселями. Они даже используют одинаковые инструменты. Меню и интерфейс пользователя у них разнятся, но конечные результате поразительно сходны.

Я часто сравниваю переход от *Photoshop* к *GIMP* с переездом из города в город. Когда вы переедете, вам потребуется время, чтобы изучить, как попасть в универсам и в театр. Но и театр, и универсам окажутся примерно такими же, как и прежние. То же и с *GIMP*. Для поиска инструментов и фильтров, выполняющих определенные задачи, нужно время и терпение, но конечный результат будет закономерен и привычен для вас.

Так почему бы, изучив *GIMP*, и вам не создавать свои шедевры? *GIMP* – не просто навороченный редактор иконок или средство ретуширования фотографий: он определенно нечто большее. Это инструмент самовыражения – выражения вашего чувства прекрасного.



› **Если вам не посчастливилось заиметь графический планшет, для рисования границ можете использовать мышь.**

Правда прекрасна, красота права

Конечно, искусство – это нечто личное: что для одного шедевр, для другого мазня. Но это не мешает вам создавать картины при помощи *GIMP* и вешать их на стены в рамах. С другой стороны, ваш супруг(а), родные люди и/или соседи по комнате могут думать иначе. Однако это тема для другого журнала.

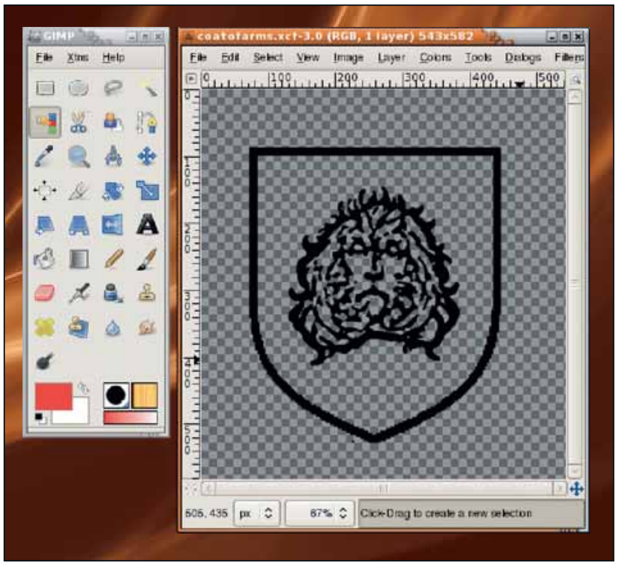
На данном уроке я покажу, как творить ради творчества. Урок прост, но он вобрал в себя основы создания ваших собственных произведений: начальный дизайн, вставку внешних изображений, раскрашивание, подсветку и тени. Раскрепостите своего внутреннего Ван Гога!

Перед тем, как запечатывать конверт, вы капнули бы на него свечой; вот и начнем с нашей собственной кляксы воска. Этот дизайн рассчитан на web, но для печати, как и во всех моих уроках, вы просто масштабируете процесс на большие изображения. На панели инструментов выберите Файл > Создать, а затем укажите шаблон 640x480. Цвета переднего плана и фона должны быть черным и белым соот-

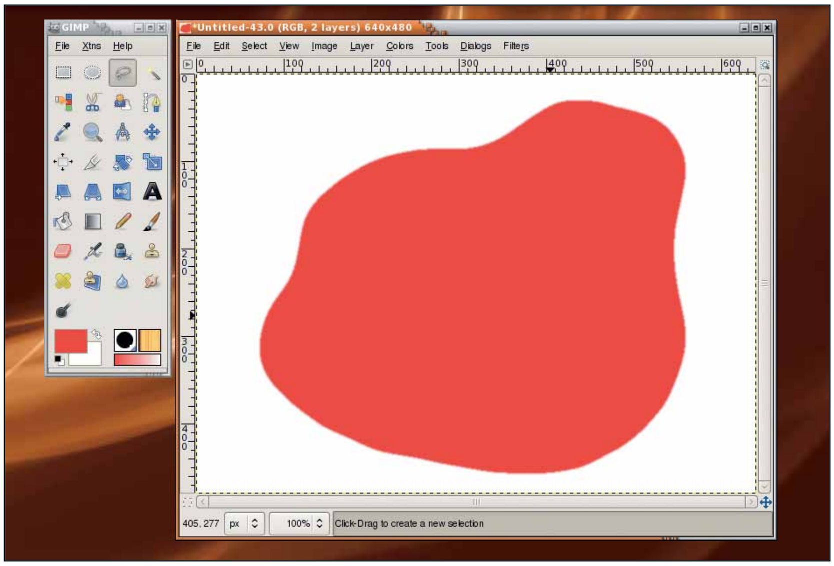
Месяц назад Мы описали разнообразие новый функций *GIMP 2.4*.

тень и текстуру

LINUX FORMAT На DVD
» Gimp 2.4.6



› Герб вообще не потребовал усилий – это просто щит (если хотите, геральдический щит) и львиная голова из коллекции рисунков.



› Сглаженное выделение не содержит гребенки на границе.

ветственно, то есть фон сперва будет белым. Если это не так, щелкните на стандартных цветовых квадратах (слева внизу области цвета). Добавьте в окно прозрачный слой (Слой > Создать слой) и назовите его **Клякса**. Выберите инструмент Свободное выделение (на панели инструментов выглядит как лассо) и нарисуйте границы кляксы в окне изображения.

Вероятно, граница получится немного зазубренная; перед тем, как заливать кляксу цветом, хотелось бы ее сгладить. Для этого воспользуемся инструментом Искажения выделения (Выделение > Исказить). Установите значение Порога в **127**, Направление и Зернистость в **1** и Сглаживание – в **35**. Следует выбрать обе опции: и Горизонтальное, и Вертикальное. Растушуйте выделение на две точки растра (Выделение > Растушевать).

Выберите цвет

Щелкните на квадрате Цвет переднего плана в панели инструментов для изменения цвета. В открывшемся диалоге установите в полях R, G и B – **227**, **47** и **47** соответственно. Этот светло-красный цвет может не годиться для вашего проекта: я его взял, потому что он хорошо смотрится в журнале. Мы изменим цвет на ваш вкус в конце урока. Щелкните на кнопке OK и примените изменения.

Перенесите квадрат Цвет переднего плана из панели инструментов в выделение. Для этого вы также можете использовать инструмент Заливка (ведро), но перенос с панели инструментов экономит несколько щелчков мыши. Снимите выделение (Ctrl+Shift+A или Выделение > Снять). Здесь мы проверим, нет ли зазубренных границ. Если выполненное нами ранее сглаживание не убрало все зазубрины, нажмите четыре раза Ctrl+Z, чтобы вернуться к начальной границе. Затем откройте диалог Искажения выделения вновь и установите более высокое значение параметра сглаживания, Повторите шаги растушевки, заливки и снятия выделения.

Клякса пока не похожа на восковую, потому что нет глубины. Мы скоро добавим ее, но перед этим нам следует подготовить штамп. Им может быть что угодно, но для эффекта старины мы создадим герб. Выберите какой-нибудь несложный, поскольку мы впоследствии его размоем, и сложный рисунок будет трудно распознать. Я создал свой собственный из щита и львиной головы.

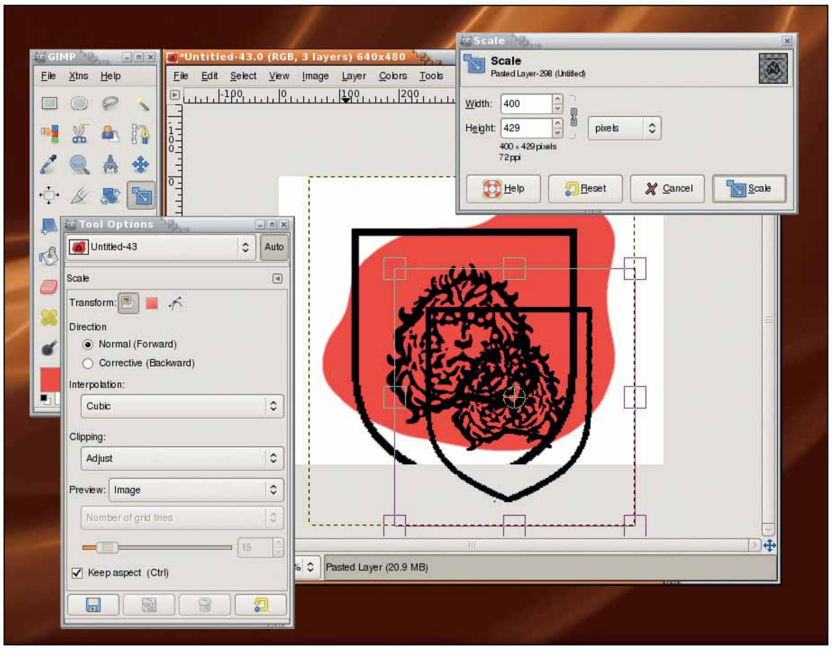
Создайте свое оружие

Контур щита и голова льва были взяты из имеющейся библиотеки картинок. Оба были объединены в одно изображение и обесцвечены. Затем я применил подстройку Кривых изображения, для повышения резкости краев каждой линии. Результат хорош для головы льва, но осталось несколько зазубрин на очертаниях щита, которые я решил не трогать, потому что это придает легкое впечатление, что штамп был вырезан вручную.

Чтобы было можно использовать штамп в этом виде, герб должен иметь прозрачные области, так что я выбрал фоновый цвет при помощи инструмента Выделение по цвету и удалил его (Ctrl+X). Скопируйте изображение герба (Ctrl+C) и вставьте его (Ctrl+V) в изображение кляксы в виде нового слоя (Слой > Создать слой после вставки) и назовите **Герб**. Вставленный слой, скорее всего, будет не того же размера, что и штамп, и попадет не туда. В этом примере штамп слишком велик, и его нужно повернуть. Выберите на панели инструмент Масштаб (он выглядит как малый квадрат со стрелкой в сторону большого). В диалоге Параметры инструмента установите галочку Сохранить пропорции, чтобы не перекосить изображение при масштабировании. Теперь »

Пара слов о терминологии

Встроенная и находящаяся на сайте справка *GIMP* всеобъемлюща, но выбор терминологии в этих документах не всегда идеален. (Я определенно не голосовал за GIMP как имя программы, поскольку для всех, знакомых с *Криминальным Чтивом*, это неудачный акроним). Однако я думаю, что в нашей серии уроков мне следует использовать ту же терминологию, что и в документации *GIMP*, ради большей пользы для читателей. Там, где терминология не очевидна, я постараюсь ее разъяснить.

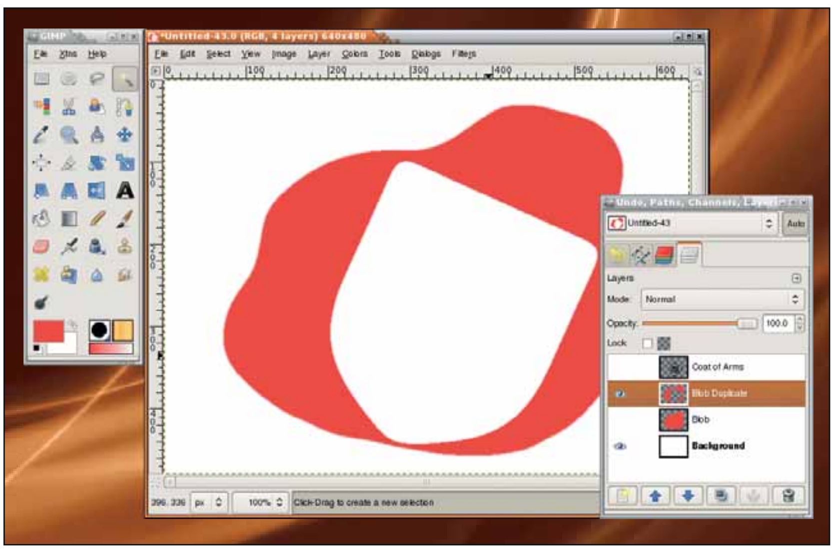


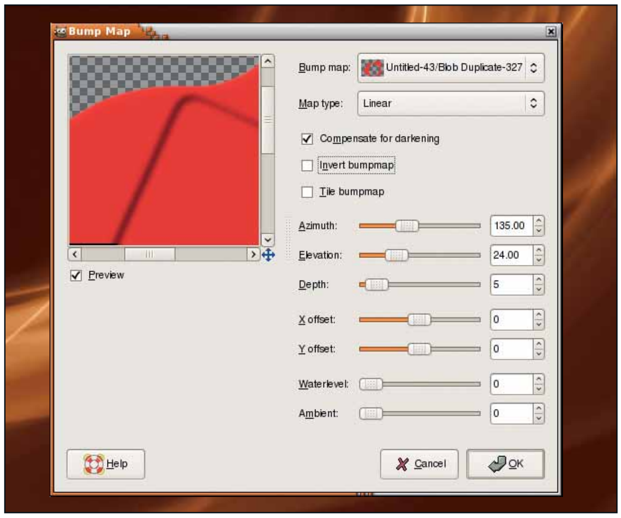
› После вращения и перемещения слоя герба может потребоваться изменить его размер.

» щелкните в окне изображения. По краям и углам слоя появятся маркеры перемещения. Щелкните и потяните один из них для изменения размера слоя, и когда герб примет требуемый размер, нажмите Enter. Используйте инструмент Перемещение для переноса герба в центр кляксы. На рисунке вверху вы можете видеть, что я сначала уменьшил масштаб отображения, чтобы лучше видеть все маркеры слоя герба. Выберите на панели инструмент Вращение и поверните герб так, чтобы он находился внутри цветной кляксы и соответствовал ее размерам. При необходимости повторите вращение и масштабирование. Установите размер слоя по размеру изображения (Слой > Слой к размеру изображения).

Для преобразования герба в оттиск на кляксе сначала надо его размыть (Фильтры > Размывание > Гауссово размывание) на три пикселя. Затем используем его для создания рельефа на цветном слое кляксы. Слой герба все еще должен быть активным (щелкните на нем в диалоге Слои, если это не так). Выберите на панели инструмент Выделение связанных областей (он напоминает тлеющую спичку) и щелкните в прозрачной области этого слоя (в данном примере это будет любая область, не включающая львиную голову или щит). Создастся выделение, состоящее из всех прозрачных областей за границами щита в гербе. Инвертируйте выделение (Выделение > Инвертировать), чтобы выбрать форму герба.

› Чтобы увидеть эффект контура, отключите видимость слоев Герб и Клякса в диалоге Слои.





› Первоначальный штамп просто применяет границы щита.

Хорошо бы сгладить и это выделение, скруглив острые углы. Почему? Потому что при вдавливании штампа в воск оттиск немного подплывает по краям. Сгладив выделение, можно имитировать этот эффект. Откройте диалог Искажение выделения вновь и примените его к нашему выделению.

Выберите слой кляксы в диалоге Слои и щелкните на кнопке дублирования внизу этого диалога. Назовите слой **Клякса-копия**. Выделение все еще должно оставаться на месте. Нажмите Ctrl+X, чтобы вырезать выделение в форме герба из слоя **Клякса-копия**.

Добавляем глубину

Теперь пора придать изображению глубину. Отключите видимость слоев **Клякса** и **Герб** в диалоге Слои, затем щелкните в диалоге Слои на слое **Клякса-копия**, чтобы сделать его активным. В окне изображения примените Гауссово размывание (Фильтры > Размывание > Гауссово размывание) на 10 пикселей к слою **Клякса-копия**.

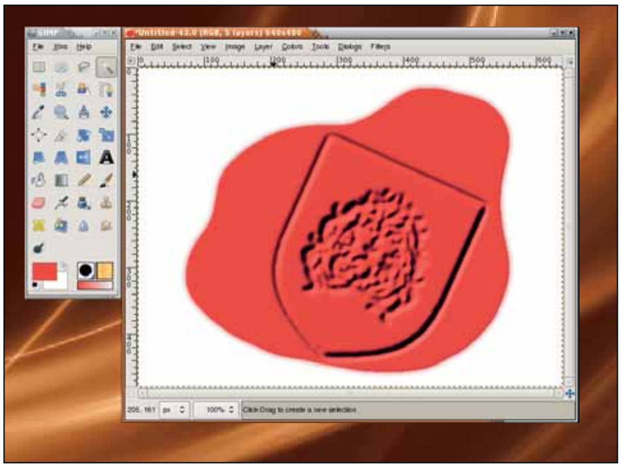
Щелкните на исходном слое **Клякса** в диалоге Слои, чтобы активировать его. Включите видимость этого слоя. Откройте фильтр Рельеф (Фильтры > Карта > Рельеф). Выберите в качестве карты рельефа слой **Клякса-копия**. Включите Компенсировать затемнение, но не активируйте опции Инвертировать или Повторять рельеф. Установите значения Азимут **135**, Возвышение **5** и Глубина **5**. Все остальные ползунки надо выставить в ноль. Примените эти настройки к исходному слою Клякса, нажав кнопку OK в диалоге.

Продублируйте слой **Герб** и назовите его **Герб-копия**. Примените Гауссово размывание на **10** пикселей к продублированному слою и инвертируйте цвета (Цвет > Инвертировать). Вновь выберите в диалоге Слои исходный слой **Клякса**. Опять откройте фильтр Рельеф, используйте те же настройки, что и ранее, но в этот раз установите опцию Инвертировать рельеф. Примените эти настройки к исходному слою **Клякса**.

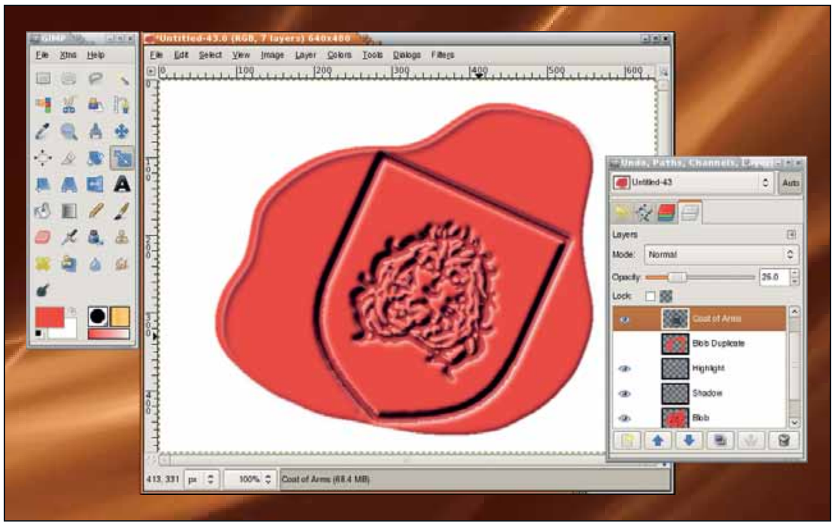
Включите видимость слоя Герб и установите его прозрачность в **25%**. Это придаст даже большую глубину эффекту штампа благодаря широким теням. Выберите в диалоге Слои слой **Герб-копия**, чтобы активировать его, и включите его видимость. Сместите этот слой (Слой > Преобразования > Сместить) на два пикселя и по X, и по Y. Выберите исходный слой Герб вновь и создайте выделение (Слой > Прозрачность > Альфа-канал > Выделение). Выберите слой **Герб-копия** и вырежьте выделение из него (Ctrl+X). Установите в диалоге Слои режим слоя-копии в Объединение зерна.

Шлифуем границы

Мы еще не все сделали. Внутри клякса воска теперь имеет глубину, но края все еще выглядят плоскими. Когда вы вдавливаете штамп, воск часто вспучивается по краям, поскольку из середины он вытесняется штампом. Сейчас мы сымитируем этот эффект. Во-первых, убедитесь,



› **Создание рельефа на основе Герба добавляет глубины в область щита, но мы не создали тени и подсветку.**



› **Эффект тени имитирует вспучивание воска после применения штампа.**

что видимость слоя **Клякса-копия** выключена. На этом уроке он нам больше не понадобится – можете, если хотите, удалить его.

Вновь продублируйте слой кляксы и назовите его **Подсветка**. Создайте выделение этого слоя (Слой > Прозрачность > Альфа-канал > Выделение) и увеличьте (Выделение > Увеличить) выделение на **2** пикселя. Заполните выделение белым, перенеся в него цвет фона с панели Цвета. Снимите выделение (Ctrl+Shift+A или Выделение > Снять). Если вы увеличили выделение недостаточно, то по краю этого слоя может получиться тонкая линия, где ранее находилось выделение. Если это ваш случай, то при помощи отмены вернитесь назад к выделению и увеличьте его еще на пару пикселей, затем вновь залейте и снимите выделение.

Сдвиньте слой на **-2** пикселя (Слой > Преобразования > Сместить) по горизонтали и вертикали. (Именно **-2**, а не **+2** пикселя.) Выберите исходный слой **Клякса** и вновь создайте из него выделение. Увеличьте его на **2** пикселя (или на то число, на которое вы увеличили предыдущее изображение). Выберите слой **Подсветка** и вырежьте из него выделение. Переместите слой подсветки на **+2** пикселя. Теперь слой выровнен, хотя вы на самом деле не можете пока видеть **Подсветку**.

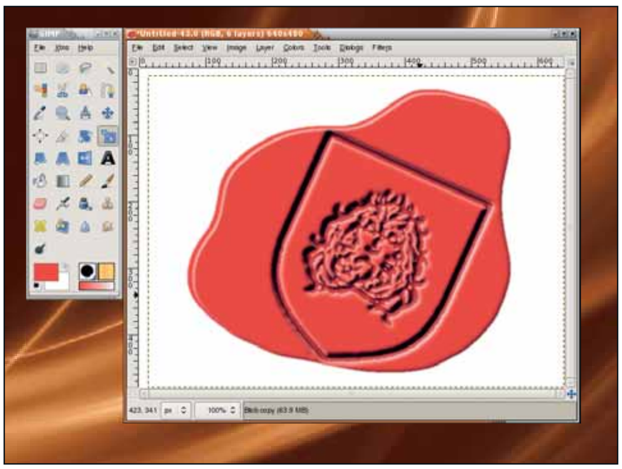
Пока слой **Подсветка** все еще активен, выберите на панели инструмент Масштаб и затем щелкните в окне изображения. Опция Сохранить пропорции должна быть включена с прошлого раза. Нажмите и перемещайте верхний левый маркер к центру изображения, пока не увидите подсветку. Подсветка должна быть размещена прямо по краю цветной кляксы. Нажмите Enter для применения изменения размера и Гауссово размывание для слоя на 3 пикселя.

Продублируйте слой **Подсветка** и назовите новый слой **Тень**. Примените к нему Гауссово размывание на 10 пикселей, затем инвертируйте цвета. Сдвиньте слой на 3 пикселя и по X, и по Y. В диалоге Слои, переместите слой ниже слоя **Подсветка**.

Завершение

Ранее я сказал, что мы начнем с яркого цвета, чтобы было легче видеть конечный результат. Теперь объединим все слои (за исключением фона), чтобы отредактировать Тон и Насыщенность.

В диалоге Слои, включите видимость для всех слоев, за исключением Фона и (если вы его еще не удалили) слоя **Клякса-копия**. Объедините видимые слои (Изображение > Объединить видимые слои), при необходимости – расширьте. Выберите диалог Тон-Насыщенность (Цвет > Тон-Насыщенность) или Тонировать (Цвет > Тонировать) и передвиньте ползунки Тон и Насыщенность по своему желанию. LXF



› **Этот шаг добавляет отражение для имитации света, но нам следует сделать еще один: добавить глубину.**

Тайны восковых печатей

В результате урока щит получился с несколько зазубренными краями по границе – из-за способа, которым я создал исходный герб, скопировав его в изображение.

Во избежание этой проблемы, попробуйте создать свой щит, используя линии для отрисовки границ, затем преобразуйте их в выделение и залейте черным. Выделите получившееся, уменьшите выделение на 3 пикселя и расширьте на 1,5 пикселя. Затем вырежьте выделение из черного щита. Тогда большинство зазубрин, обнаруженных в очертаниях щита, будет удалено.

» **Через месяц** Мы займемся более геометричным узором – сборкой из шестеренок.

Inkscape: Новое

Inkscape 0.46 – серьезное усовершенствование популярного редактора векторной графики. **Дмитрий Кирсанов** показывает, что там к чему.



Наверное, это самый продвинутый векторный графический редактор с открытым исходным кодом на всех ОС: ненавязчивый графический интерфейс, масса возможностей и поддержка открытого формата SVG делают его прекрасным выбором.

Векторные редакторы типа *Inkscape* отличаются от растровых вроде *GIMP* тем, что в них линия рассматривается как множество точек, описанных уравнением, а не как набор пикселей, формирующих картинку. Поэтому векторные редакторы берут верх в тех случаях, когда изображения нужно масштабировать для печати (например, постеры и логотипы); а растровые редакторы превосходят их при работе с фотографиями, видео и сложными изображениями.

В конце марта была выпущена версия *Inkscape 0.46*, находившаяся в разработке больше года. И какая версия! Десятки крупных и сотни мелких новых возможностей и улучшений (включая результаты весьма продуктивного Google Summer of Code 2007) обусловили серьезную переработку внутренней архитектуры.

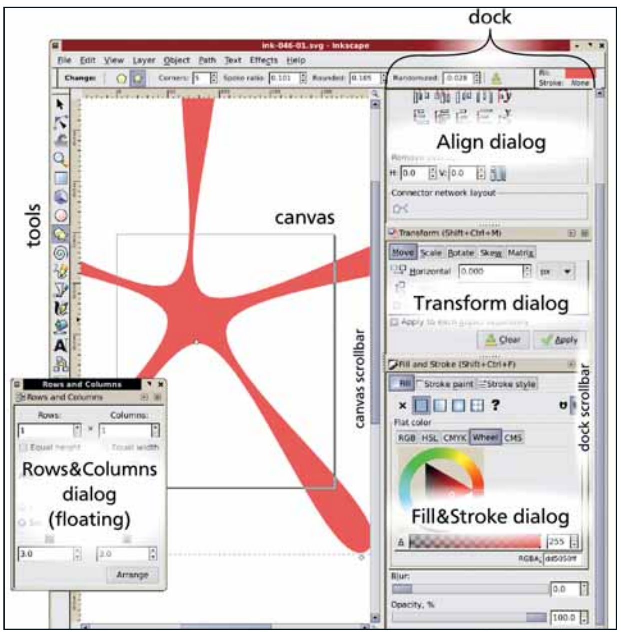
Один из показателей размера приложения – число текстовых строк в интерфейсе, а в этой версии *Inkscape* оно выросло на 40 %. Понятно, что наш короткий обзор способен только раздразнить аппетит!

Основные изменения по сравнению с Inkscape 0.45



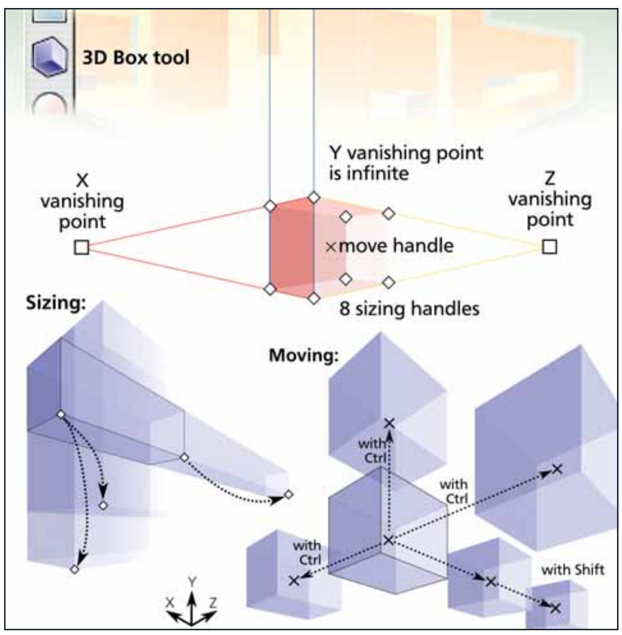
Наш эксперт

Дмитрий Кирсанов свободный дизайнер, автор книг и SVG-разработчик, а также ветеран Linux и участник проекта *Inkscape*.



1 Док для диалоговых окон

Наверное, самое заметное изменение в интерфейсе – это диалоговые окна с фиксированным местоположением. Если открыть, например, диалог Заливка и обводка, по умолчанию он появится в прокручиваемом вертикальном доке справа в главном окне. Другие диалоговые окна также помещаются туда, и с помощью мыши их можно переупорядочить и изменить их размер, а также вытащить из дока, и они станут плавающими. Положение каждого диалога и его состояние док/плавающий сохраняются между сеансами работы.



2 Инструмент Параллелепипед

В глубине души *Inkscape* – редактор двумерных изображений. Однако в нем часто рисуют трехмерные объекты. Чтобы помочь в этом, в версии *0.46* появился новый тип объекта: параллелепипед – базовый элемент перспективы, неизменной со времен Ренессанса. С ним можно создавать прямоугольники, менять их размеры и перемещать их в трехмерии, а также подгонять точки схода и линии перспективы. В других инструментах 3D рассматривается как набор из шести граней, и каждую можно выделить и преобразовать как обычный контур.



в версии 0.46

При всех новинках версии 0.46, интерфейс по умолчанию изменился мало. И это, наверное, хорошо – вам не придется переучиваться: почти все работает как прежде. Новые функции не осложняют программу, обнаруживая себя только при необходимости. В некоторых случаях – например, редактировании контрольных точек градиента прямо на холсте – в *Inkscape 0.46* просто реализовали то, что планировалось с самого начала, но не было закончено.

Важная вещь, которую не покажешь на экранных снимках – быстродействие. Как ни странно, пару лет назад, когда *Inkscape* был гораздо медленнее, большинство пользователей считали его бодрячком. Сейчас, несмотря на повышение скорости обновления экрана и уменьшение времени отклика, жалобы на медлительность и расход оперативной памяти участились. Причина проста: *Inkscape* созрел для по-настоящему крупных задач, и художники, создающие в нем сложные работы, все чаще начинают выходить из себя.

Хотя в версии 0.46 произошли некоторые желанные изменения в скорости рендеринга и отзывчивости интерфейса, настоящего прорыва здесь нет. Отрисовка стала существенно быстрее лишь в режиме отображения только контурных линий (Вид > Режим > Каркас), для чего сейчас используется API Cairo. К сожалению, Cairo еще не готов для рендеринга в обычном режиме, но этого можно реально ожидать уже в следующей версии.

И это еще не все...

Многие улучшения версии 0.46 не уместились в этой статье. Однако вот краткий перечень достойных внимания:

» Предпросмотр перед выводом на печать, если задан цветовой ICC-профиль устройства вывода (так называемый «программный оттиск»).

» Новая команда для создания маркера выделенного объекта. Сейчас легко создавать пользовательские маркеры, используя любую функцию *Inkscape* – градиенты, прозрачность, даже фильтры.

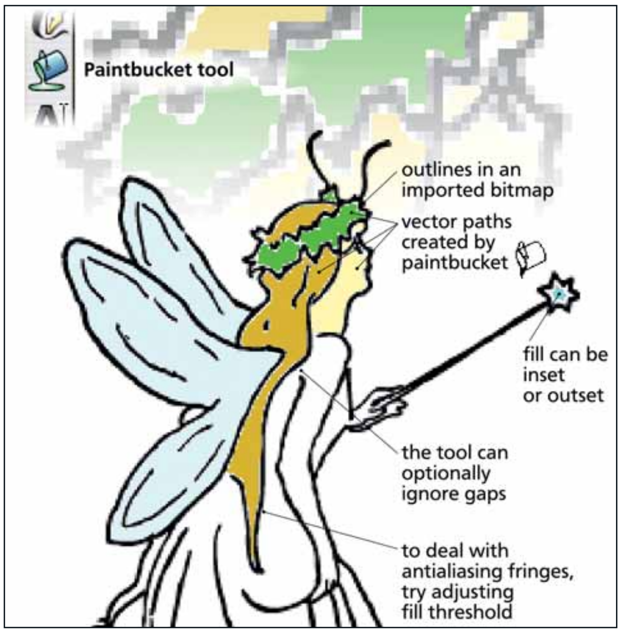
» Новые функции инструмента Каллиграфическое перо, особенно слежение за направляющей и штриховка фона с изменяемой шириной пера, упрощают создание сеток, подобных используемым в гравюрах.

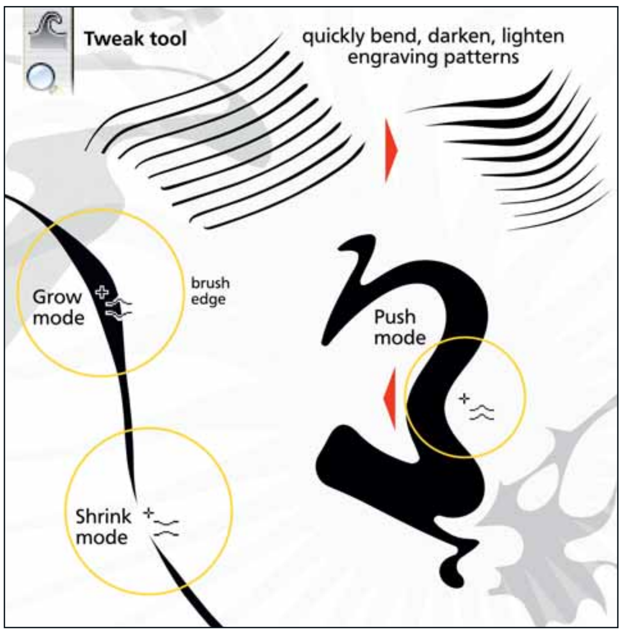
» Множество стандартных шаблонов (в полоску, в клетку, в горошек и т.д.), доступных в диалоге Заливка и обводка при переключении объекта на заливку или штрихование шаблоном.

» *UniConverter* (если он установлен в вашей системе) – программа на Python – может использоваться в *Inkscape* для импорта файлов *Corel Draw* (CDR).

» Все команды *Inkscape*, как и выделение объектов по их идентификаторам, теперь можно вызывать из командной строки, что позволяет автоматически выполнять довольно сложные сценарии (например, загрузить файл, выбрать объект, удалить его, сохранить файл и выйти).

» Перешедшие в *Inkscape* из *Adobe Illustrator* теперь могут пользоваться пробелом для панорамирования: нужно лишь включить это в настройках. После этого нажатие пробела и перемещение мышью панорамирует изображение (по умолчанию пробел, как и прежде, переключает на Селектор).





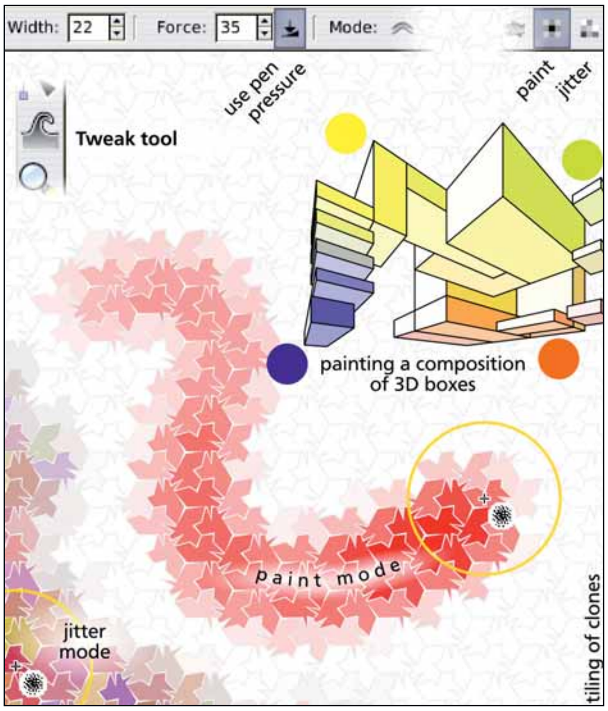
3 Инструмент заливки цветом

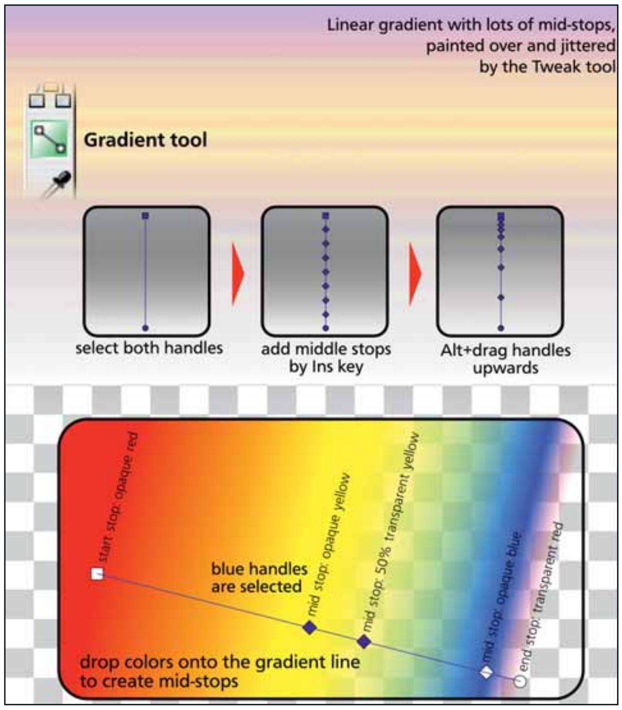
Еще один новый инструмент – Плоская заливка. Подобно большинству инструментов *Inkscape*, он работает так же, как и в любом растровом редакторе. Кликните на холсте, и инструмент создаст контур, заполняющий область цветом, начиная с указанной вами точки. При определении границ заливаемой области Заливка смотрит на то, что видно на холсте – она игнорирует полностью прозрачные фигуры, но позволяет закрасить одноцветные области в импортированном растровом рисунке.

4 Подгонка кривых

Inkscape 0.46 избавляет нас от монополии мощного, но ограниченного инструмента для редактирования кривых – Узлов. С новым инструментом Корректор можно слепить любое количество кривых, нажимая, утолщая, отбрасывая, притягивая и огрубляя. Больше незачем беспокоиться об узлах и точках-рукоятках – просто примените широкую мягкую кисть этого инструмента (её размер и нажим регулируются), и выбранные кривые плавно и естественно изменят свою форму, будто сделанные из глины.

»





» **5**

Настройка цветов

Корректор не только изменяет форму кривых – два из его режимов позволяют легко редактировать цвета объектов. Ранее в векторных редакторах можно было раскрасить объекты, только выделив их, выбрав цвет и, затем, назначив его. Версия *0.46* все изменила. Теперь можно понемногу «втирать» цвет в объекты, так что в режиме рисования они смешиваются, а в режиме перебора – выбираются случайным образом. Закрашивание множества небольших объектов (типа узора из клонированных фигур) очень похоже на рисование мягкой кистью в растровом редакторе (с планшетом, инструмент может использовать давление пера для установки нажима).

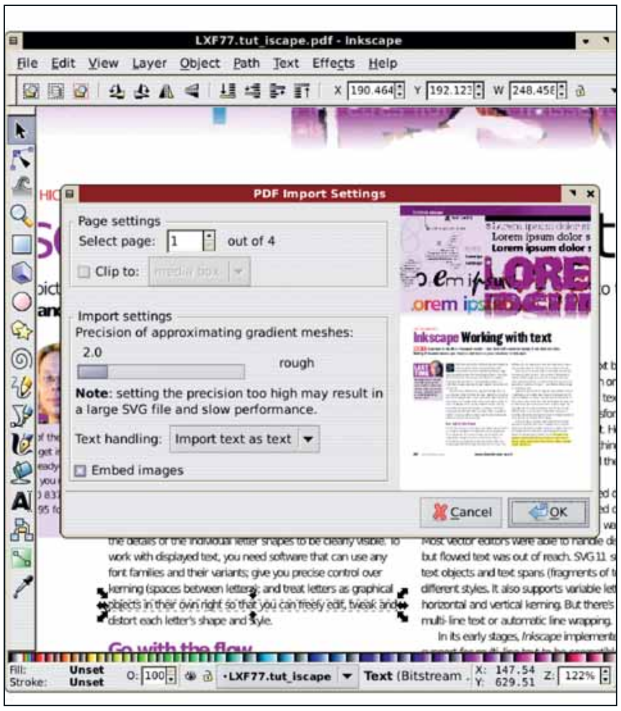
6

Промежуточные точки градиента

Начиная с версии *0.42* в Inkscape можно редактировать градиенты на холсте, перетаскивая рукоятки и назначая им цвета. Прежде свои собственные рукоятки были только на конечных точках градиента. Можно было создать и промежуточные, но только в неуклюжем Редакторе градиентов. В версии *0.46* у промежуточных точек тоже есть рукоятки; их можно создать, перетащив цвет из палитры на градиент или сделав двойной щелчок левой кнопкой мыши на линии градиента. Другие удобства: выделение нескольких рукояток одновременно, вставка промежуточных точек клавишей Ins, создание рукояток с помощью Alt+мышь и рисование/рандомизация цветов многих промежуточных точках с помощью Корректора.

Скорая помощь

Чтобы изменить цвет выделения, не открывая диалоговых окон, захватите образец заливки или штриха с левого конца строки статуса и перетащите его в область документа; далее руководствуйтесь сообщениями в строке статуса.





7

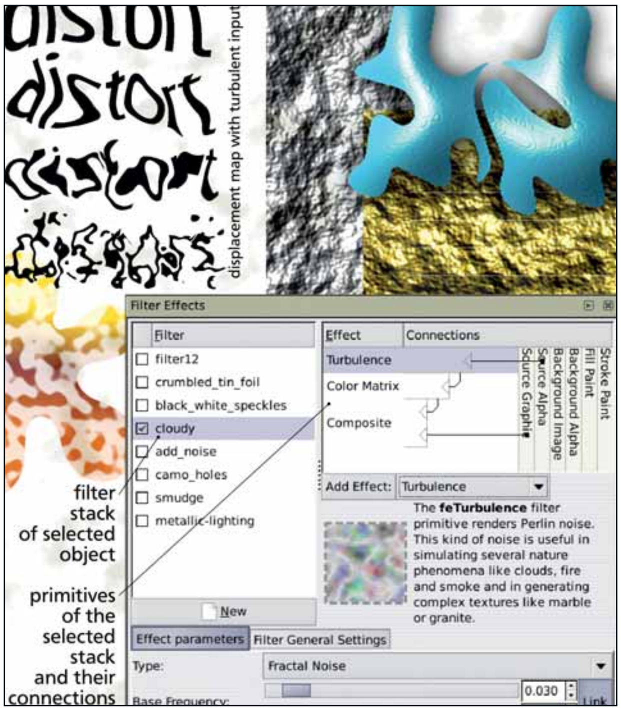
Импорт PDF

Новая реализация импорта PDF позволяет выбирать страницы в многостраничных файлах и сохраняет все векторные и растровые рисунки и даже редактируемый текст. Так как «родной» формат AI *Adobe Illustrator* (с версии 9.0) также основан на PDF, в *Inkscape* теперь можно открывать любые файлы AI. Добавление модуля экспорта на основе Cairo делает версию *0.46* полноценным решением для редактирования одностраничных PDF.

8

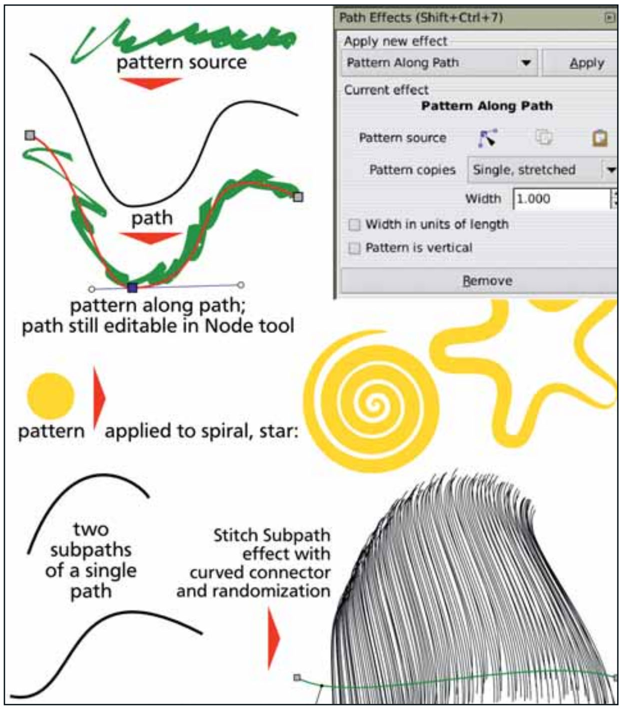
Импорт/экспорт открытого контента с OCAL

Будучи открытым продуктом, *Inkscape* поддерживает и открытый контент. С некоторого времени в диалоге Метаданные можно подписать файл SVG и назначить ему открытую лицензию. В версии *0.46* можно импортировать и экспортировать картинки с сайта **www.openclipart.org**. В диалоге импорта OCAL можно искать по тэгам и просматривать миниатюры, выбранное изображение вставляется прямо в документ.



9 Фильтры SVG

Фильтры – одна из тех вещей, которые отличают SVG от других векторных форматов. Это, по сути, растровая операция, применяемая к обрабатываемому изображению без утраты его векторной природы. В версии *0.45* поддерживался только один фильтр: Гауссово размывание. В версии *0.46* доступно большинство фильтров, определенных в формате SVG, и есть диалоговое окно, позволяющее создавать стеки эффектов, давая в результате множество сложных эффектов.

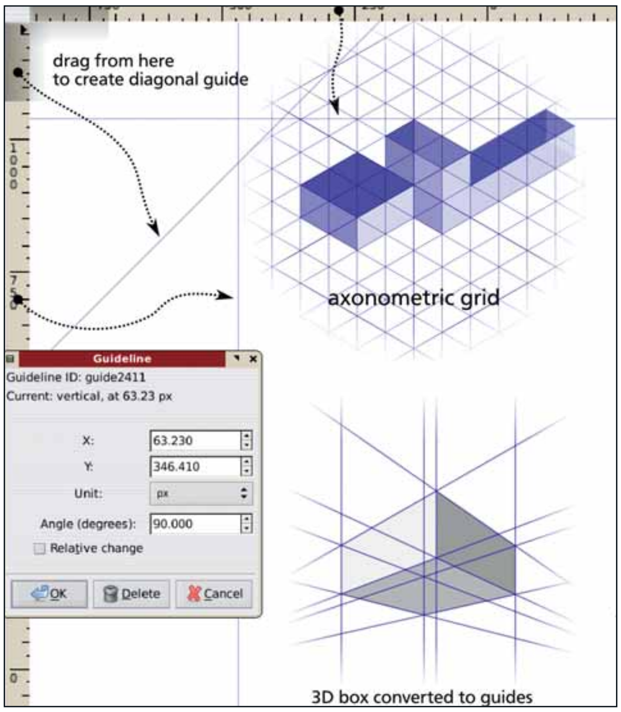


10 Эффекты динамических контуров

Эффект динамического контура (Life Path Effect, LPE) – это неразрушающее преобразование исходной кривой (т.е. формы фигуры, а не ее стиля). Кривая с LPE может совсем не походить на оригинал, но последний сохраняется, и его еще можно редактировать инструментом Узлы. Удобнее всего эффекты По контуру, который изгибает одну кривую вдоль другой, и Сшивка субконтуров, соединяющий вложенные кривые сетью линий.

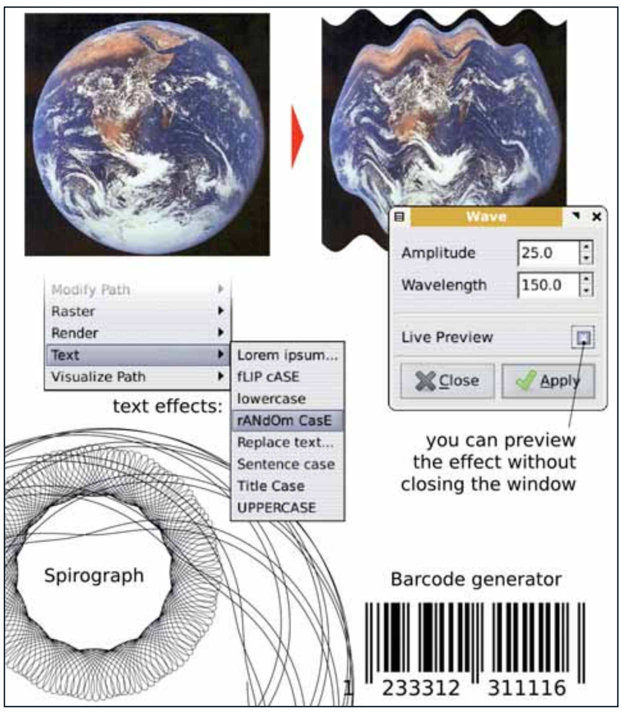
Скорая помощь

Чтобы применить эффект динамического контура некоторого объекта к нескольким кривым или фигурам, скопируйте исходный объект (Ctrl+C), выберите целевой и вызовите эффект Вставить контур (Ctrl+7).



11 Сетка, линейки и привязка

Многие используют *Inkscape* в основном для технических иллюстраций, где все должно быть выровнено идеально. Таким пользователям наверняка понравятся новые наклонные линейки, аксонометрическая сетка и возможность преобразовывать объекты в направляющие или наличие более одной сетки в документе. Поведение линий привязки в версии *0.46* значительно улучшилось – теперь привязка стала быстрее и логичнее.



12 Новые расширения

Как всегда, вместе с новой версией *Inkscape* появляется и масса новых расширений (скрипты Python для обработки документа). Есть растровые эффекты (например, Обесцветить, Рельеф и Повысить резкость), эффекты для текстовых объектов (изменение регистра, поиск и замена) и т.д. Для расширений в версии *0.46* также поддерживается XSLT, и XSLT-фильтр поставляется для экспорта/импорта в формат XAML. LXF

Скорая помощь

Для применения стека фильтров одного объекта к другим, скопируйте исходный объект (Ctrl+C), выберите целевые и нажмите Ctrl+Shift+V. Так можно скопировать и вставить любые фильтры из примеров, поставляемых с *Inkscape*.

Arduino: Светодиодный

В завершение нашей серии **Грэм Моррисон** соберет датчик освещенности.

Что нам нужно:

- Arduino
- Макетная плата
- Пять резисторов по 1 кОм
- Датчик освещенности

Наш эксперт

Грэм Моррисон
Разработчик свободного ПО, фанат KDE и энтузиаст электронной музыки. Он обсуждал синтезаторы не с одним ведущим членом сообщества FOSS.

Это последний учебник по Arduino в нашей серии. Прибавим проекту жизни: вставим датчик освещенности. Из названия ясно, что он должен как-то реагировать на свет, но эти датчики не так умны, как кажутся. По сути, датчик освещенности – это переменный резистор, и его также часто называют «фоторезистором» [light dependent resistor, LDR], что прекрасно описывает принцип действия. Фоторезистор можно использовать в самых разных задачах, от измерения освещенности помещения до построения сети защитных датчиков и лазеров вокруг короны Св. Эдуарда в Лондонском Тауэре [используемой при коронации британских монархов, – *прим. ред.*]. А главное, датчик освещенности обычно имеется в стартовом наборе Arduino, и для построения чего-то полезного вам больше почти ничего не нужно.

Шаг 1: Сборка

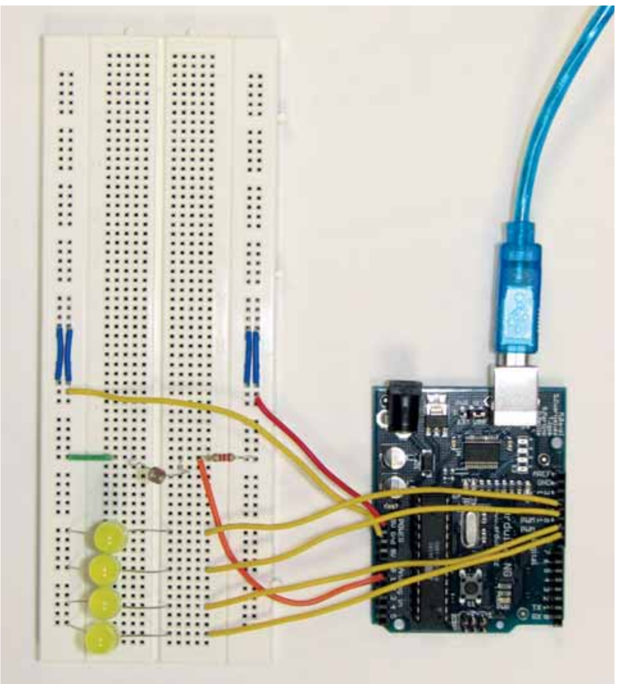
Если вы, например, прикрутите датчик освещенности к простой цепи со светодиодом, то обнаружите, что яркость последнего меняется в зависимости от количества прямого света, падающего на датчик, Вы также обнаружите, что эта яркость немного колеблется, независимо от освещенности датчика, потому что цепь «плывет», и возникающая небольшая электрическая интерференция влияет на сопротивление. Решением будет добавить резистор в 1 кОм между источником энергии и датчиком освещенности. Это стабилизирует цепь, остановив интерференцию, влияющую на сопротивление датчика.

Вы можете попытаться сделать это сами. Поместите датчик освещенности на вашу макетную плату – так, чтобы два его контакта были по обе стороны центральной канавки. Соедините одну сторону через резистор с 5-В выходом Arduino и соедините проводом ту же линию с Аналоговым входом 0. Другую сторону канавки макетной платы просто подключите к GND (Земля) на Arduino. Наконец, вставьте длинную ножку светодиода прямо в Цифровой порт 13 на Arduino, а короткую ножку – в ближайший разъем GND. Вот код, делающий рабочей эту конструкцию:

```
int ledPin = 13;
int analogInput = 0;
int aInput;
void setup(){
 pinMode(ledPin, OUTPUT);
 pinMode(analogInput, INPUT);
 beginSerial(9600);
}
void loop(){
 aInput = analogRead(analogInput);
 digitalWrite(ledPin, HIGH);
 delay(aInput);
 digitalWrite(ledPin, LOW);
 delay(aInput);
 printInteger(aInput);
 printByte(10);
}
```

Вы должны бы уже привыкнуть к синтаксису, и понять, что при этом происходит, несложно. Мы считываем аналоговые данные с датчика освещенности и используем их как задержку мигания светодиода на плате. Чем ярче свет, падающий на датчик, тем меньше задержка между вспышками светодиода. Как мы изучили в прошлом месяце, аналоговые значения варьируются в диапазоне от 0 до 1023, что может дать задержку вспышек от более одной секунды до практически нулевой. Но вы найдете, что большинству датчиков нужен прямой солнечный свет, чтобы задержки стали меньше 150 миллисекунд (единицы измерения, используемые командой 'delay'), и при искусственном освещении различия заметить труднее. Значит, нам нужен способ увидеть значения, считываемые с датчика освещенности, чтобы мы могли улучшить нашу программу для работы с меньшими значениями. Вот это и делают строки beginSerial(9600), printInteger(aInput) и printByte(10). Они открывают последовательное соединение с компьютером и начинают передавать значения с датчика освещенности. Чтобы увидеть значение, нужно просто открыть Serial Monitor в IDE Arduino на вашем компьютере. Вы будете видеть каждое значение в новой строке (результат работы команды printByte), и сможете отслеживать читаемые с датчика освещенности числа, отражающие количество света. В условиях светлого офиса наш датчик показывал около 800 и сбрасывался до 960, когда мы закрывали его пальцем. Учтите, датчик редко имеет линейную зависимость отклика, и при уменьшении освещенности вдвое вы не должны ожидать такого же уменьшения сопротивления.

Разобравшись с выводом с датчика, мы можем применить сенсор еще где-нибудь. Используем массив светодиодов как столбик графика, где число горящих светодиодов представляет значение освещенности,



› На вид сложновато, но этот проект датчика освещенности очень легко реализуется и пригодится в различных видах приложений.

Месяц назад Мы достали всю семью, злоупотребляя битьем в пьезобарабан.

светометр



попадающей на датчик. Можете использовать до 13 отдельных светодиодов – столько, сколько позволяют цифровые входы Arduino, но для демонстрации основных принципов нам хватит четырех. Каждый светодиод требует присоединения его длинной ножки к отдельному цифровому входу, в большинстве случаев через 1-кОм резистор (это зависит от типа используемых светодиодов). Короткая ножка подключается к «земле». Используем различные контакты для каждого светодиода, с общей шиной, присоединенной к GND, и проводом к цифровому входу Arduino.

Шаг 2: Программируем

Вот и все нужные нам дополнительные подключения. Но исходный код программы требует чуть больше работы. Начнем с верхней части кода, добавив следующие определения 'define' и изменив переменную ledPin:

```
#define MAX_LED 4
int ledPin[MAX_LED]={12, 11,10,9};
```

Вместо того, чтобы вручную инициализировать и обрабатывать каждый подключенный к Arduino светодиод, мы создали простой массив, куда записали выходной контакт для каждого соединения (ledPin). Мы не только сэкономили место на данной странице, но и облегчили легкое перемещение по данным. Это решение также более гибкое. Если вы захотите добавить или удалить светодиоды с Arduino, вам будет только нужно изменить значение MAX_LED и удалить назначенные контакты из массива. Но перед тем, как мы сможем их зажечь, сперва необходимо инициализировать каждый используемый контакт, в функции setup заменив строку pinMode(ledPin, OUTPUT) на следующую:

```
for (int i=0; i<MAX_LED; i++){
    pinMode(ledPin[i], OUTPUT);
}
```

Этот простой цикл for показывает вам, как легко сконфигурировать каждый светодиод в массиве, и мы в дальнейшем применим эту технику еще дважды в новой функции с именем barGraph. Она выполняет работу по зажиганию требуемого числа светодиодов, в зависимости от переданного функции входного значения. Например, если оно равно **3**, будут задействованы первые три светодиода в нашем массиве. Используем эту функцию для показа освещенности, определяемой датчиком:

```
void barGraph(int count) {
  for (int i=0; i<MAX_LED; i++){
    digitalWrite(ledPin[i], LOW);
  }
  for (int i=0; i<count; i++){
    digitalWrite(ledPin[i], HIGH);
  }
}
```

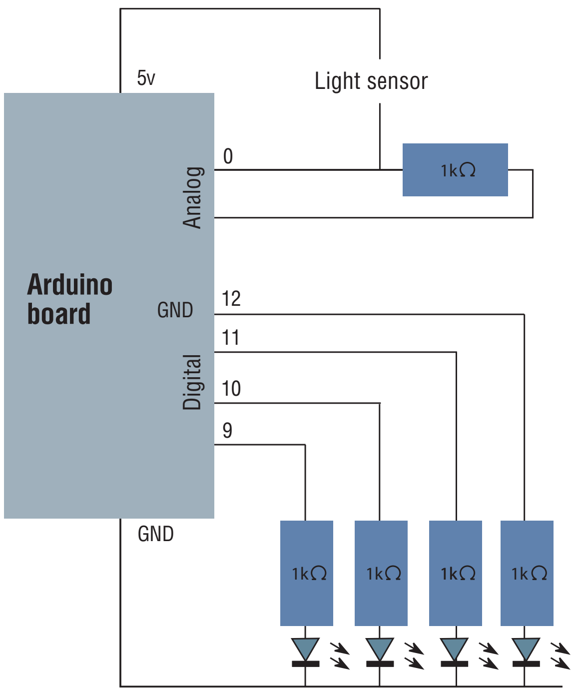
Тут сперва выключаются все светодиоды, а потом они по очереди включаются, в зависимости от значения переменной 'count'. Наконец, в логику программы осталось добавить команду, ответственную за вызов функции barGraph с различными значениями. Мы используем серию условий if для снятия различных пороговых значений освещенности. Следующие значения работали у нас, но вам может потребоваться использовать Serial Console в IDE Arduino для подбора значений, более подходящих к вашей установке. Если значения, читаемые с датчика освещенности, менее чем **840**, мы зажигаем все четыре светодиода. Это значение освещенности в нашем офисе. Если вам повезет испытывать установку на природе летним солнечным днем, то более подходящим значением будет **150**. Каждый последующий диапазон для 3-х, 2-х, 1-го и 0 светодиодов является произвольным отклонением от стартового значения. Вы можете использовать более чем четыре светодиода, для этого вам нужно будет добавить соответствующие диапазоны в условия 'if'.

```
aInput = analogRead(analogInput);
 if (aInput < 840)
   barGraph(4);
 else if (aInput < 900)
   barGraph(3);
 else if (aInput < 940)
   barGraph(2);
 else if (aInput < 980)
   barGraph(1);
 else
   barGraph(0);
delay (10);
```

Скомпилировав и послав эту программу на ваш Arduino, вы должны увидеть набор светодиодов, откликающихся в зависимости от уровня света, попадающего на сенсор. Темнота все их выключит, а прямое освещение все зажжет. Проект готов. Потом вы сможете много чего добавить, чтобы извлечь из него пользу. Можно применить его в системах домашней автоматизации – например, включения или выключения устройств при наступлении темноты. Лучшей стартовой площадкой будет Arduino Sketchbook – онлайн-репозиторий всех вещей Arduino, а также полезный форум, полных энтузиастов вроде вас. Все это можно найти на главном сайте Arduino: www.arduino.cc. Сообщите нам, если изваяете что-нибудь клевое. LXF

Скорая помощь

Если вы пытаетесь установить или обновить IDE Arduino и он начнет генерировать длинные и странные сообщения об ошибке компиляции, убедитесь, что у вас также установлен пакет *avr-libc*.



Ядро: о сущности

Ядро – незаменимая часть ПО в центре вашего компьютера, но что оно, собственно, делает? **Д-р Крис Браун** вскроет его и заглянет внутрь.





Наш эксперт

Д-р Крис Браун
Пользователь Unix с более чем двадцатипятилетним стажем. Его компания Interactive Digital Learning предоставляет курсы, консультации и материалы для обучения по Linux. Он также специализируется на системах электронного обучения.

Мой верный Оксфордский словарь определяет ядро как «мягкую, съедобную часть ореха», но есть и второе значение: «Центральная или очень важная часть чего-либо». (Кстати, первое определение породило термин 'Оболочка', что на языке Linux означает командный интерпретатор.) Для тех, кто имеет смутное представление, что именно делает ядро, приведем немножко теории.

Ядро (kernel) – это кусочек ПО, который, грубо говоря, представляет собой слой между оборудованием и приложениями, запущенными на компьютере. Если придерживаться строгой терминологии, 'Linux' означает только ядро – программу, которую Линус Торвальдс написал в начале 90-х годов. Все остальные части, которые вы можете найти в составе дистрибутива Linux – оболочка *Bash*, графическое окружение KDE, web-браузеры, *X*-сервер, *Tux Racer* и многое другое – это всего лишь приложения, запускаемые поверх ядра Linux, и они, подчеркнем это, не являются частью самой операционной системы. Чтобы оценить долю ядра, скажем, что свежеустановленный RHEL5 занимает около 2,5 ГБ на жестком диске (понятно, что точное число зависит от вашего выбора пакетов). Из них само ядро со всеми модулями занимает 47 МБ, или около 2 %.

Внутри ядра

Но что все-таки делает ядро? Диаграмма на рис. 1 (через страницу) показывает это. Ядро предоставляет приложениям свои службы, которые можно запустить через множество специальных точек входа – системных вызовов. С точки зрения программиста, это выглядит как простой вызов функции, хотя на самом деле системный вызов влечет за собой явное переключение режима процессора из пространства пользователя (user space) в пространство ядра (kernel space). Все вместе системные вызовы образуют «виртуальную машину Linux», которую можно представить как абстракцию нижележащего аппаратного обеспечения.

Одна из самых понятных абстракций, предоставляемых ядром, это файловая система. Для примера приведем маленькую программу (написанную на C), которая открывает файл и копирует его содержимое в стандартный вывод:

```
#include <fcntl.h>
int main()
{
  int fd, count; char buf[1000];
  fd=open("mydata", O_RDONLY);
  count = read(fd, buf, 1000);
  write(1, buf, count);
  close(fd);
}
```

Здесь вы видите пример четырех системных вызовов – open, read, write и close. Не вдавайтесь в детали синтаксиса: сейчас это не главное. Главное вот что: через эти (и некоторые другие) системные вызовы ядро Linux предоставляет нам иллюзию «файла» – ряда байтов с данными, который имеет имя – и защищает вас от нижележащих деталей: дорожек, секторов и головок со списком свободных блоков, которые вам бы пришлось учитывать при непосредственном доступе к оборудованию. Вот что мы подразумеваем под абстракцией.

Как видно из рис. 1, ядро выполняет сложную работу по созданию такой абстракции, когда сама файловая система может быть сохранена во множестве различных форматов, на локальных устройствах хранения, таких как жесткие диски, CD или USB-брелки – а то и на удаленной системе через сетевые протоколы вроде NFS или CIFS. Может даже иметься еще и дополнительный слой отображения устройств для поддержки логических томов или RAID. Слой виртуальной файловой системы (VFS) внутри ядра представляет эти нижележащие формы устройств хранения как коллекцию файлов внутри единой иерархической файловой системы.

За кулисами

Файловая система – одна из самых понятных абстракций, предоставляемых ядром. Но некоторые функции не видимы так явно. Например, ядро отвечает за планирование процессов. В любой момент времени существуют различные процессы, ожидающие запуска. Планировщик ядра предоставляет каждому часть процессорного времени, и если смотреть в течение нескольких секунд, у вас складывается иллюзия работы нескольких программ одновременно. Вот небольшая программа на C:

```
#include <stdlib.h>
main()
{
  if (fork()) {
    write(1, "Parent\n", 7);
```

Месяц назад Мы показали, как использовать проект Linux Terminal Server.

модулей

```
    wait(0);
    exit(0);
}
  else {
    write(1, "Child\n", 6);
    exit(0);
  }
 }
```

Она создает новый процесс; и исходный процесс (родительский), и новые процессы (дочерние) пишут сообщение в стандартный вывод и завершаются. Снова, забудьте про синтаксис. Просто отметьте, что системные вызовы fork(), exit() и wait() отвечают за создание, завершение и синхронизацию процессов соответственно. Эти элегантно-простые вызовы скрывают стоящую за ними сложность управления и планирования процессов.

Еще более скрытые, даже от программистов, функции ядра – это управление памятью. Каждый запускаемый процесс уверен, что имеет доступное только ему одному адресное пространство (допустимый набор адресов памяти). На самом деле он разделяет ОЗУ компьютера с множеством других процессов, и если памяти в системе не хватает, то часть его адресного пространства переедет на диск в область подкачки [swap]. Другой аспект управления памятью – защита адресного пространства одного процесса от доступа других – необходимое условие для сохранения целостности многозадачной операционной системы [оно реализуется аппаратно, – прим. ред.].

Ядро также осуществляет работу с сетевыми протоколами, такими как IP, TCP и UDP, предоставляющими соединение компьютер-компьютер и процесс-процесс через сеть. Снова вернемся к иллюзиям. TCP создает иллюзию постоянного соединения между двумя процессами – как кусок кабеля, соединяющий два телефона – хотя на самом деле никакого постоянного соединения не существует. Заметим также, что конкретные прикладные протоколы, такие как FTP, DNS или HTTP, применяемые в пользовательских программах, не являются частью ядра.

Linux (как и его «предок» Unix) имеет хорошую репутацию по части безопасности. Она достигается тем, что ядро отслеживает ID пользователя и группы для каждого запущенного процесса и использует их для ответа «да/нет» при каждом обращении процесса к ресурсу (например, открытию файла для записи), проверяя права доступа. Именно эта модель контроля доступа обеспечивает основную безопасность Linux-систем.

Наконец (сразу приношу извинения программистам, написавшим те части ядра, которые я не перечислил), ядро сопровождается большой коллекцией модулей, способных общаться с аппаратными устройствами на языке низкого уровня: считать сектор с диска, получить пакет с сетевой карты и так далее. Их также иногда называют драйверами устройств.

Модульное ядро

Теперь, получив представление о том, что делает ядро, давайте кратко рассмотрим его физическую организацию. Ранние версии ядра Linux были монолитными – это когда все части и кусочки были статически собраны в один (довольно большой) исполняемый файл. Современные ядра Linux, напротив, модульные: большая часть их функций содержится в модулях, динамически загружаемых в ядро. Это позволяет сохранять размер основной части ядра небольшим и делает возможным подключение и замену модулей в запущенном ядре без перезагрузки.

Достаем исходники ядра

Чтобы добыть исходные коды ядра, не надо быть суперхакером. Вы можете просто установить пакет с ними, предоставляемый вашим дистрибутивом, так же, как устанавливаете другие пакеты.

Например, в моем Ubuntu я запускал команду:

```
# apt-get install linux-source
```

Альтернатива – попытаться зайти на www.kernel.org и загрузить архив с исходными текстами (он может называться как-то вроде **linux-2.6.25.6.tar.bz2**) в **/tmp**. Распакуйте его в **/usr/src**:

```
# cd /usr/src
# tar xjf /tmp/linux-2.6.24.2.tar.bz2
```

В обоих случаях вы в итоге получите исходные коды ядра в поддиректории **/usr/src** с именем, содержащим номер версии; обычно что то вроде **/usr/src/linux-source-2.6.25**.

Основная часть ядра помещается в память во время загрузки системы, из файла в директории **/boot** с именем вида **vmlinuz-ВЕРСИЯ_ЯДРА**, где **ВЕРСИЯ_ЯДРА** – это, сами понимаете, номер версии текущего ядра. (Чтобы узнать, какая версия ядра используется у вас, выполните команду uname -r.) Модули ядра находятся в директории **/lib/modules/ВЕРСИЯ_ЯДРА**. Все эти кусочки были скопированы на место при установке ядра.

Управляем модулями

Как правило, Linux разбирается со своими модулями без нашей помощи, но на случай необходимости есть команды для обследования и управления модулями вручную. Например, чтобы найти, какие модули в настоящее время загружены в ядро, используйте *lsmod*. Вот пример его вывода:

```
# lsmod
pcspkr          4224 0
hci_usb         18204 2
psmouse         38920 0
bluetooth       55908 7 rfcomm,l2cap,hci_usb
yenta_socket    27532 5
rsrc_nonstatic 14080 1 yenta_socket
isofs           36284 0
```

Поля этого вывода содержат имя модуля, его размер, счетчик его использования и список зависимых от него модулей. Счетчик использования важен для защиты модуля от выгрузки при его активности. Linux может удалить модуль, только когда его счетчик использования равен нулю.

Модули можно загружать и выгружать вручную, используя *modprobe*. (На самом деле, работа выполняется двумя низкоуровневыми командами, *insmod* и *rmmod*, но *modbrobe* проще в использовании, потому что автоматически разрешает зависимости модулей.) Например, вывод *lsmod* на нашей машине показал загруженный модуль с именем *isofs*, счетчик использования которого равен нулю и от него не зависят другие модули. (*isofs* – это модуль поддержки файловой системы ISO, используемой на CD). Ядро с удовольствием избавится от такого модуля при помощи команды:

```
# modprobe -r isofs
```

Isofs больше не показывается в выводе *lsmod*, и, что ценно, ядро стало потреблять на 36284 байта меньше памяти. Если вы вставите CD диск и он смонтируется автоматически, ядро также автоматически подгрузит модуль *isofs* и увеличит счетчик его использования до 1. Попытка удалить модуль вам на сей раз не удастся, так как он используется: »

```
# modprobe -r isofs
FATAL: Module isofs is in use.
```

lsmod просто показывает модули, загруженные в данный момент; а вот *modprobe -l* выведет список всех доступных модулей, то есть все модули, находящиеся в **/lib/modules/ВЕРСИЯ_ЯДРА**; будьте готовы к длинному перечню.

В реальности, вам обычно незачем загружать модули вручную с помощью *modprobe*, но если придется, можно заодно передавать в модуль параметры, через командную строку *modprobe*. Например, так:

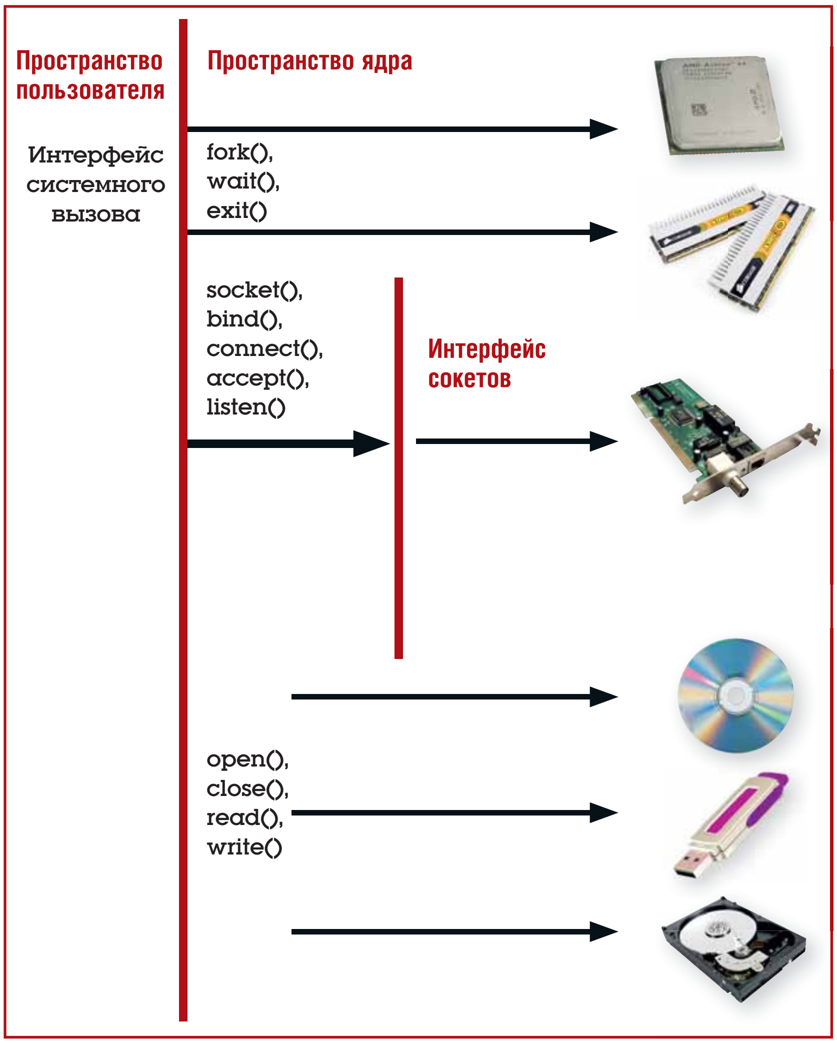
```
# modprobe usbcore blinkenlights=1
```

Термин blinkenlights («мигалка») отнюдь не выдуман – это реальный параметр модуля *usbcore*.

Сложность здесь заключается в знании, какие параметры имеет модуль. Можете использовать звонок другу или помощь зала, но лучше обратиться к команде *modinfo*: она выводит список различной информации о модуле.

Вот пример информации, выводимой для модуля *snd-hda-intel* (в сокращении):

```
# modinfo snd-hda-intel
filename:       /lib/modules/2.6.20-16-generic/kernel/sound/pci/
hda/snd-hda-intel.ko
description: Intel HDA driver
license:        GPL
srcversion: A3552B2DF3A932D88FFC00C
 alias:       pci:v000010DEd0000055Dsv*sd*bc*sc*i*
 alias:       pci:v000010DEd0000055Csv*sd*bc*sc*i*
 depends:     snd-pcm,snd-page-alloc,snd-hda-codec,snd
 vermagic:    2.6.20-16-generic SMP mod_unload 586
 parm:        index:Index value for Intel HD audio interface. (int)
 parm:        id:ID string for Intel HD audio interface. (charp)
 parm:        model:Use the given board model. (charp)
 parm:        position_fix:Fix DMA pointer (0 = auto, 1 = none, 2 =
POSBUF 3 = FIFO size). (int)
 parm:        probe_mask:Bitmask to probe codecs (default = -1).
(int)
 parm:        single_cmd:Use single command to communicate
with codecs (for debugging only). (bool)
 parm:        enable_msi:Enable Message Signaled Interrupt
(MSI) (int)
 parm:        enable:bool
```



› **Рис. 1. Ядро использует системные вызовы 'read' и 'write' для представления абстракции вашего оборудования.**

Интересующие нас строки начинаются с parm: – они показывают параметры, принимаемые модулем. Их описание, мягко выражаясь, сжатое. Для охоты за дальнейшей информацией установите исходные коды ядра, затем найдите директорию, имеющую имя вроде **/usr/src/ВЕРСИЯ_ЯДРА/Documentation**. В ней содержится множество интересных вещей: например, файл **/usr/src/ВЕРСИЯ_ЯДРА/Documentation/sound/alsa/ALSA-Configuration.txt** описывает параметры, распознаваемые многими звуковыми модулями ALSA. Файл **/usr/src/ВЕРСИЯ_ЯДРА/Documentation/kernel-parameters.txt** тоже полезен.

Пример необходимости передачи параметров в модуль недавно обсуждался на одном из форумов Ubuntu (см. **https://help.ubuntu.com/community/HdaIntelSoundHowto**): говорилось, что модуль *snd-hda-intel* нуждается в небольшой помощи для правильного управления звуковым оборудованием и иногда «зависает» во время загрузки при старте системы. Часть исправления заключалась в применении к модулю опции probe_mask=1. Итак, если вы загружаете модуль вручную, напечатайте:

```
# modprobe snd-hda-intel probe_mask=1
```

А лучше поместить эту строку в файл **/etc/modprobe.conf**, следующим образом:

```
options snd-hda-intel probe_mask=1
```

Она велит *modprobe* включать опцию probe_mask=1 при каждой загрузке модуля *snd-hda-intel*. Некоторые существующие дистрибутивы Linux разбивают эту информацию по множеству файлов директории **/etc/modprobe.d**, а не сосредоточивают ее всю в **modprobe.conf**.

Файловая система /proc

Ядро Linux также экспонирует множество информации через файловую систему **/proc**. Чтобы получить понятие о **/proc**, мы должны расширить наше представление о файлах и рассматривать их не как хранилище информации на жестком диске, CD или карте памяти, а как любую информацию, к которой можно получить доступ через стандартные системные вызовы, типа открыть/читать/записать/закрыть, рассмотренные нами ранее, или через обычные программы вроде *cat* или *less*. «Файл» в **/proc** полностью является плодом воображения ядра и предоставляет возможность просмотра данных множества внутренних структур последнего.

Фактически, многие справочные утилиты Linux представляют в человеко-читаемой форме информацию, найденную в файлах **/proc**. Например, в **/proc/modules** содержится список текущих загруженных модулей, странным образом напоминающий вывод *lsmod*. Аналогично, **/proc/meminfo** содержит прорву деталей о текущем состоянии виртуальной памяти системы, а утилиты типа *vmstat* и *top* предоставят часть этой информации в (чуть) более внятном формате. Другой пример – содержимое **/proc/net/arp**, которое показывает текущее состояние кэша ARP; в командной строке *arp -a* покажет похожую информацию.

Особо интересны «файлы» в **/proc/sys**. Например установки в **/proc/sys/net/ipv4/ip_forward** говорят ядру, должен ли компьютер пересылать адресованные не ему IP-датаграммы – т.е. выполнять функцию шлюза. В данном случае, ядро сообщает, что эта возможность выключена:

```
# cat /proc/sys/net/ipv4/ip_forward
0
```

А еще интереснее будет новость, что можно также и писать в эти файлы. Продолжая наш пример:

Стоит ли трудов тюнинг производительности?

Первым автомобилем моего отца был Wolseley 1500 с номером 49 RNU – уж и не знаю, как я упомнил эти подробности. Так или иначе, отец любил эксперименты и тратил часы на мелкие доводки вроде угла зажигания или качества смешения топлива. Время от времени он откручивал свечи и корректировал зазоры. Выкрутив свечи, он подавал топливо в цилиндры, как часть некого таинственного процесса внутреннего орошения. После этого выхлоп превращался в убедительные густые облака черного дыма.



Но вот беда: у него не было никаких объективных способов измерить, к чему привели его улучшения. Он скрупулезно вел записи о купленном бензине и километраже, рассчитывая расход топлива до нескольких десятичных знаков, и у него был специальный холм, куда он поднимался на третьей передаче, чтобы «посмотреть, как пошло», но все это нельзя было назвать научной методикой.

Многие системные администраторы Linux находятся в аналогичном положении. Они знают, что есть множество параметров, которые можно изменить и повысить производительность, но слабо представляют, что делает большинство из них, и не имеют хорошего способа для измерения производительности. Итак, мой вам совет: если вы не знаете, что вы делаете, и/или нет способа измерить результат, оставьте эти параметры в покое!

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

включит IP-переадресацию в запущенном ядре.

Вместо использования *cat* или echo для исследования и модификации настроек в **/proc/sys** можете использовать команду *sysctl*.

```
# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
```

эквивалентно

```
# cat /proc/sys/net/ipv4/ip_forward
0
```

тогда как

```
# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

аналогично

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Заметим, что пути, применяемые вами в *sysctl*, используют для разделения компонентов точку (.) вместо привычного правого слэша (/), и что этот путь вычисляется относительно **/proc/sys**.

Помните, что настройки, изменяемые таким образом, будут в силе только на текущий запуск ядра – они не сохранятся при перезагрузке. Чтобы сделать их постоянными, поместите их в файл **/etc/sysctl.conf**. При загрузке системы *sysctl* должна будет автоматически применить любые настройки, найденные ею в этом файле.

Строка в **/etc/sysctl.conf** будет выглядеть так:

```
net.ipv4.ip_forward=1
```

Тюнинг производительности

Записываемые параметры в **/proc/sys** породили целую субкультуру тюнинга производительности Linux. Лично я считаю это излишним, но вот вам несколько примеров на пробу.

Инструкция по установке *Oracle 10g* (**www.oracle.com/technology/obe/obe10gdb/install/linuxpreinst/linuxpreinst.htm**) просит вас указать несколько параметров, включая

```
kernel.shmmax=2147483648
```

устанавливающий максимально возможный разделяемый сегмент памяти в 2 ГБ. (Разделяемая память представляет собой механизм межпроцессной коммуникации, когда сегмент памяти является видимым в адресных пространствах множества процессов.)

Руководство IBM 'Redpaper' по настройке производительности и тюнингу Linux (**www.redbooks.ibm.com/abstracts/redp4285.html**) всячески советует применять параметры в **/proc/sys**, например:

```
vm.swappiness=100
```

Этот параметр, очевидно, управляет интенсивностью сброса страниц памяти на диск.

Некоторые параметры можно подправить для усиления безопасности. Сайт Боба Кромвеля (**http://cromwell-intl.com/security/security-stack-hardening.html**) содержит несколько хороших примеров, включая такой:

```
net.ipv4.icmp_echo_ignore_broadcasts=1
```

Он предписывает ядру не отвечать на широковещательные ICMP-запросы (ping'и), делая вашу сеть менее чувствительной к одной из DoS-атак (отказ в обслуживани), известной также как Smurf-атака.

В другом примере,

```
net.ipv4.conf.all.rp_filter=1
```

ядру велено усилить уровень проверки, также называемый входной или выходной фильтрацией. Суть этого заключается в отбраковке пакетов, если IP-адрес отправителя или получателя в заголовке пакета не имеют смысла при его рассмотрении в свете использованного физического интерфейса.

А есть ли документация по всем этим параметрам ? Да – команда

```
# sysctl -a
```

покажет вам их имена и текущие значения. Это длинный список, но по нему не догадаешься, что параметры делают. Есть ли другие источники? Оказывается, издательство O'Reilly издало книгу, написанную Оливье Доделем [Olivier Daudel], под названием *«/proc et /sys»*. Oui, mes amis, она на французском, и русского перевода нет. Другая полезная ссылка – *Red Hat Enterprise Linux Reference Guide*, руководство, содержащее главы о рассматриваемом объекте. Вы можете скачать его с **www.redhat.com/docs/manuals/enterprise**. Полезной книгой о ядре Linux *является Ядро Linux*. 3-е издание Бовета и Чезати (O'Reilly/BHV), но сразу предупредим, что она о внутренностях ядра, и будет полезна скорее будущим разработчикам и студентам компьютерных специальностей, чем системным администраторам.

Также возможно сконфигурировать и собрать собственное ядро. Для этого прочтите великолепный учебник Нейла Ботвика в LXF89 или *Linux Kernel in a Nutshell Грега Кроа-Хартмана* [Greg Kroah-Hartman] издательства O'Reilly, заголовок которой содержит очаровательную, но, по-видимому, непредумышленную игру слов [по-англ. in a nutshell буквально значит «в скорлупе ореха», а в переносном смысле – «вкратце», – *прим. ред.*]. Но, конечно, чтобы собирать ядро, надо быть крепким орешком. LXF

» **Через месяц** Внедрим новости в жизнь, скрестив RSS и Festival!



Обмен

ЧАСТЬ 6 За последние годы D-Bus стал стандартом де-факто для межпроцессного взаимодействия на рабочем столе Linux. Что здесь может предложить *Qt*? Разбирается **Андрей Боровский**.

В отличие от возможностей *Qt 4.x*, с которыми мы познакомились ранее, появившийся в *Qt 4.2* модуль *QtDBus* присутствует только в версиях *Qt*, предназначенных для Unix-систем. Объясняется это, конечно же, тем, что полнофункциональной версии D-Bus для Windows не существует (хотя работы в этом направлении ведутся). Мы уже рассматривали архитектуру шины D-Bus в LXF99, и там была высказана мысль, что программу-клиент D-Bus, которая только обращается к сервисам другого приложения, нетрудно написать даже на «голом» C. С сервером дела обстоят сложнее: он должен уметь обрабатывать сообщения D-Bus, поступающие асинхронно. *QtDBus* упрощает решение этой задачи настолько, насколько это вообще возможно. В качестве демонстрации возможностей *QtDBus* мы напишем программу-сервер, которая будет предоставлять доступ к буферу обмена X-Window консольным приложениям. Во времена господства MS-DOS и Windows 3.1 существовал DOSовский текстовый редактор (название я, к сожалению, уже не помню), который, будучи запущен в DOS-окне Windows, мог обмениваться данными с буфером обмена Windows. Делалось это с помощью какого-то хитроумного прерывания, и мне тогда казалось, что это очень круто. Нечто подобное, только для X, мы сейчас и напишем.

Минуточку внимания!

Берясь за *QtDBus*, имейте в виду: фирменная документация по нему зияет пробелами (чего не скажешь о других модулях). Это тем более странно, что *QtDBus* появился в *Qt 4* уже давно. Нет, описания классов модуля *QtDBus* присутствуют, и довольно подробны, но вот, например, демонстрационное приложение, описанное на странице http://doc.trolltech.com/4.4/qdbusadaptorexample.html, не только не работает, но даже и не компилируется (текст этого примера не менялся, кстати, с версии *Qt 4.2*).

Программа-сервер

Каждое приложение-сервер D-Bus должно предоставлять как минимум один интерфейс (то есть описание набора методов, к которым можно обратиться при помощи D-Bus из другого приложения). Интерфейсы D-Bus предоставляются объектами. И если на уровне D-Bus объект – понятие скорее условное, то при программировании сервера D-Bus в *Qt 4* самый простой способ объявить интерфейс заключается в том, чтобы реализовать класс, предоставляющий требуемый интерфейс, а затем создать объект этого класса и зарегистрировать его в качестве поставщика интерфейса. Рассмотрим объект, реализующий интерфейс нашего приложения-сервера (полный текст вы найдете на диске в архиве **clipboarviewer.tar.gz**):

```
#include <QApplication>
#include <QtCore>
#include <QtDBus>
#include <QClipboard>

class QCBAdapter: public QDBusAbstractAdaptor
{
    Q_OBJECT
    Q_CLASSINFO("D-Bus Interface", "DBus.Manager.QClipboard")
    Q_PROPERTY(QString cbContent READ content WRITE setContent)

 private:
    QApplication *app;

 public:
    QCBAdapter(QApplication *application)
       : QDBusAbstractAdaptor(application)
    {
      cb =  QApplication::clipboard();
    }
    Q_INVOKABLE QString content()
    {
            printf("Запрос содержимого буфера обмена\n");
            return cb->text();
    }

   Q_INVOKABLE void setContent(const QString &newContent)
   {
           printf("Содержимое буфера обмена изменено\n");
      cb->setText(newContent);
   }

public slots:
   Q_NOREPLY void emptyClipboard()
   {
           cb->clear();
   }
```

» **Месяц назад** Мы посмотрели, как можно использовать скриптовый движок Qt в ваших программах.

сообщениями

```
private:
        QClipboard * cb;
};
```

Помимо прочего, мы включаем в исходный текст заголовочный файл **QtDBus**. Кстати, не забудьте добавить строку

```
QT += D-Bus
```

в **.pro**-файл вашего приложения. Основой всех классов, реализующих интерфейсы D-Bus, должен быть класс QDBusAbstractAdaptor. От него и происходит наш класс QCBAdapter. Макрос Q_CLASSINFO() позволяет определить имя экспортируемого интерфейса в формате, принятом в D-Bus (один класс может экспортировать несколько интерфейсов). Интерфейс нашего класса состоит из свойства cbContent, позволяющего получить доступ к текстовому содержимому буфера обмена (если таковое имеется) и вспомогательных функций: content() – для чтения содержимого буфера обмена и setContent() – для его записи. Обратите внимание на то, что оба метода помечены макросом Q_INVOKABLE. Это необходимо для того, чтобы программы, которые не умеют выполнять маршаллинг свойств интерфейса, могли обратиться к ним напрямую. Остальные элементы класса QCBAdapter не должны вызвать вопросов. Получив указатель на глобальный объект класса QClipboard, мы манипулируем содержимым буфера обмена с помощью методов QClipboard::setText() и clear(). Наш интерфейс предназначен для чтения и записи только текстовой информации, но консольной программе, скорее всего, большего и не потребуется.

Перейдем теперь к функции main(), которая создает и регистрирует объект, реализующий интерфейс.

```
#include "QCBAdapter.h"

int main(int argc, char **argv)
{
        QApplication app(argc, argv);
        QCBAdapter * adapter = new QCBAdapter(&app);
        QDBusConnection connection = QDBusConnection::connectToBus(QDBusConnection::SessionBus, "DBus.Manager.QClipboard");
        if (connection.isConnected())
                printf("Соединение установлено\n");
        if (!connection.registerService("DBus.Manager.QClipboard")) {
                printf("Не могу зарегистрировать сервис\n");
                exit(1);
        }
        if (!connection.registerObject("/QClipboard", adapter, QDBusConnection::ExportAllContents)) {
                printf("Не могу зарегистрировать объект\n");
                exit(1);
        }
    app.exec();
}
```

Прежде всего, обратите внимание на то, что в нашей программе отсутствуют графические элементы. Они нам и не нужны – приложение ClipboardViewer должно выполняться как сервер. При этом следует учесть, что хотя программа ClipboardViewer не создает никаких окон, она связывается с X-сервером (иначе как бы она смогла получить доступ к буферу обмена?), а значит, является полноценным приложением X. После того как мы создали объекты классов QApplication и QCBAdapter, необходимо выполнить соединение с демоном D-Bus. Напомню, что в рамках архитектуры D-Bus любые две программы могут создать собственную шину D-Bus, однако существуют две стандартные шины – системная System Bus и пользовательская Session Bus. Наш сервер буфера обмена проще всего подключить к пользовательской шине.

Соединение Qt-программы с D-Bus инкапсулируется объектом класса QDBusConnection. Мы создаем его с помощью статического метода connectToBus(). Вторым аргументом данного метода должно быть имя соединения, которое программы-клиенты будут использовать для подключения к нашему серверу. Напомню, что по своей структуре оно напоминает доменное имя Интернета, и если бы у нашей программы был свой сайт, его имя можно было бы включить в имя соединения. Многие имена соединений, не связанные с сайтами, все равно начинаются с префиксов org.* или com.*, но, как мы покажем на практике, следование этому правилу вовсе не является обязательным.

Установив соединение с демоном, мы можем зарегистрировать имя сервиса, предоставляемого нашим сервером. Оно имеет ту же структуру, что и имя соединения, и может совпадать с ним (обратите внимание, что имя сервиса должно совпадать с именем интерфейса, указанным в объявлении класса адаптера с помощью Q_CLASSINFO()). Как вы уже догадались, регистрация сервиса выполняется методом registerService() объекта класса QDBusConnection.

Нам осталось лишь зарегистрировать объект, реализующий сервис – это выполняется методом registerObject(). Первый аргумент метода – путь к объекту, который используется системой D-Bus для его идентификации. Выбор /QClipboard – чисто произвольный, можно указать любое другое значение, например, /the/longest/path/to/the/object. Последний аргумент метода registerObject() определяет, какие элементы класса адаптера войдут в описание интерфейса D-Bus и станут доступными удаленному приложению. По сути дела, мы сталкиваемся здесь с той же проблемой видимости элементов класса за пределами приложения, что и в случае со сценариями Qt. Константа QDBusConnection::ExportAllContents делает класс-адаптер максимально видимым для удаленных приложений, но даже при ее использовании другим программам будет доступно не так уж и много. Например, программа-клиент D-Bus сможет обращаться только к методам, помеченным как Q_INVOKABLE. Если все прошло успешно, нам остается запустить цикл обработки сообщений нашего приложения.

Клиент на языке C

Настало время написать программу-клиент для взаимодействия с сервером. Чтобы продемонстрировать универсальный характер D-Bus, напишем сначала программу на языке C. Она естественно, не сможет использовать средства *Qt*:

```
#include <stdio.h>
#include <dbus/dbus.h>
int main (int argc, char **argv)
{
 DBusConnection * connection;
 DBusError error;
```

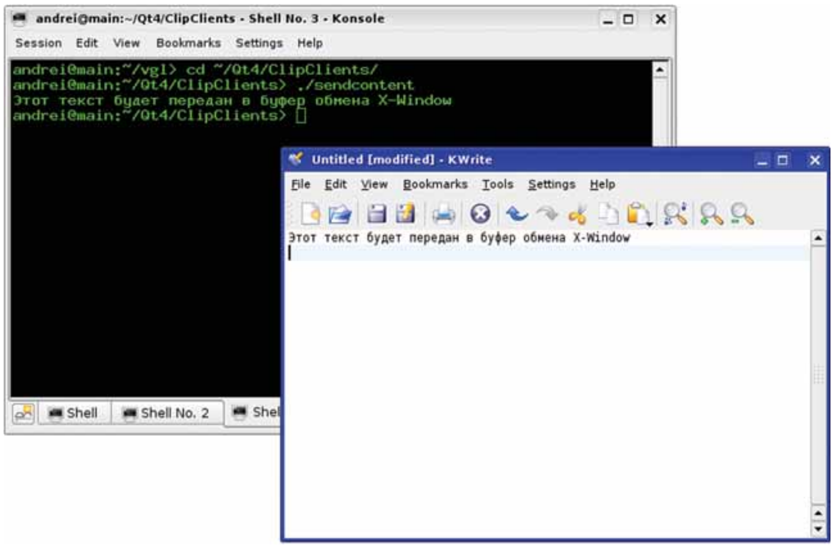
```
DBusMessage *call;
char * text = "Этот текст будет передан в буфер обмена X-Window";
dbus_error_init(&error);
connection = DBus_bus_get(DBUS_BUS_SESSION, &error);
if (!connection) {
 printf("Не могу установить соединение\n", error.message);
 dbus_error_free(&error);
 return 1;
}
call = DBus_message_new_method_call("DBus.Manager.QClipboard", "/QClipboard", "DBus.Manager.QClipboard", "setContent");
dbus_message_append_args (call, DBus_TYPE_STRING, &text, DBus_TYPE_INVALID);
if (!dbus_connection_send(connection, call, NULL)) {
 printf("Не могу отправить сообщение\n");
}
dbus_connection_flush(connection);
printf("%s\n", text);
dbus_message_unref(call);
dbus_connection_unref(connection);
return 0;
}
```

Программа, исходный текст которой вы найдете в файле **sendcontent.c** в архиве **clipclients.tar.gz**, записывает строку текста в буфер обмена X и распечатывает ее же в окне консоли. Для компиляции программы следует использовать команду

```
gcc sendcontent.c `pkg-config --cflags dbus-1` `pkg-config --libs dbus-1` -o sendcontent
```

Разбирать подробно работу программы *sendcontent* мы не будем, так как она не имеет отношения к *Qt*, и, к тому же, фактически представляет собой сокращенный вариант примера из статьи, опубликованной в LXF99. Заметим только, что функция dbus_message_new_method_call() использует имя соединения, путь к объекту и имя сервиса, заданные нами в программе-сервере.

Запустите программу *ClipBoardViewer* в окне консоли. Затем в другом окне запустите программу *sendcontent*. После этого вы сможете вставить переданную консольной программой *sendcontent* строку в буфер обмена X-Window (**рис. 1**).



› Рис. 1. В наше прогрессивное время никакие «хитроумные прерывания» не нужны – D-Bus запросто передаст текст из консоли в окно *Kwrite*.

Базовый клиент D-Bus

Теперь посмотрим, как можно решить ту же задачу базовыми средствами *Qt*. Мы напишем настоящую консольную *Qt*-программу. У нее нет объекта QApplication, нет доступа к объекту QClipboard, а значит, для передачи данных в буфер обмена X ей придется воспользоваться нашим сервером. Ниже приводится текст программы sendlines (возможно, это первая консольная программа *Qt*, которую вы видите).

```
#include <stdio.h>
#include <QtCore>
#include <QtDBus>
int main(int argc, char **argv)
{
		QCoreApplication app(argc, argv);
		QDBusConnection connection = QDBusConnection::connectToBus(QDBusConnection::SessionBus, "example.sendlines");
		if (connection.isConnected())
				printf("Соединение установлено\n");
		else {
				printf("Не могу установить соединение\n");
				return 1;
		}
		while(true) {
				QString string = "";
				int ch;
				while ((ch = getchar()) != '\n')
						string += ch;
				QDBusMessage msg = QDBusMessage::createMethodCall("DBus.Manager.QClipboard", "/QClipboard", "DBus.Manager.QClipboard", "setContent");
				QList<QVariant> args;
				args.append(QVariant(string));
				msg.setArguments(args);
				connection.send(msg);
				QCoreApplication::processEvents();
		}
}
```

Модуль *QtGUI* включается по умолчанию во все проекты *Qt 4*, поэтому при создании консольного приложения Qt в файл **.pro**, помимо строки

```
QT += D-Bus
```

следует добавить также

```
QT –= gui
```

Теперь компоненты графического приложения будут недоступны нашему проекту. Вместо класса QApplication мы используем QCoreApplication. Вместо вызова метода exec() объекта QApplication мы «вручную» создаем цикл обработки событий с помощью статического метода QCoreApplication::processEvents(). Но самое интересное в нашей программе, конечно, не это.

Прежде всего, нам опять понадобится объект QDBusConnection, который мы создаем точно так же, как и в программе ClipboardViewer. Далее наша программа входит в цикл ввода строк (из которого ее можно вывести только с помощью комбинации Ctrl-C). После того, как пользователь завершает ввод очередной строки, создается объект msg класса QDBusMessage – он инкапсулирует сообщение D-Bus. Мы создаем объект-сообщение типа «вызов метода» с помощью статического метода createMethodCall(). Аргументами метода createMethodCall() должны быть соответственно имя сервиса, путь к объекту, имя интерфейса и имя вызываемого метода.

Те, кто читал статью, посвященную D-Bus, должны помнить, что с сообщениями этого типа обычно связываются дополнительные данные, представляющие собой значения аргументов вызываемого метода удаленного объекта. Они добавляются в объект msg с помощью метода setArguments(). Его аргументом должен быть список QList, состоящий из элементов типа QVariant. Мы создаем последний с одним-единственным элементом (соответствующим единственному аргументу метода setContent), вызываем метод setArguments(), после чего отправляем созданное сообщение с помощью метода send() объекта класса QDBusConnection. Теперь мы можем передать в буфер обмена X-Window серию строк, что можно проследить, например, с помощью апплета Klipper (**рис. 2**).



› Рис. 2. Базовый клиент D-Bus может передать в буфер обмена сразу несколько строк.

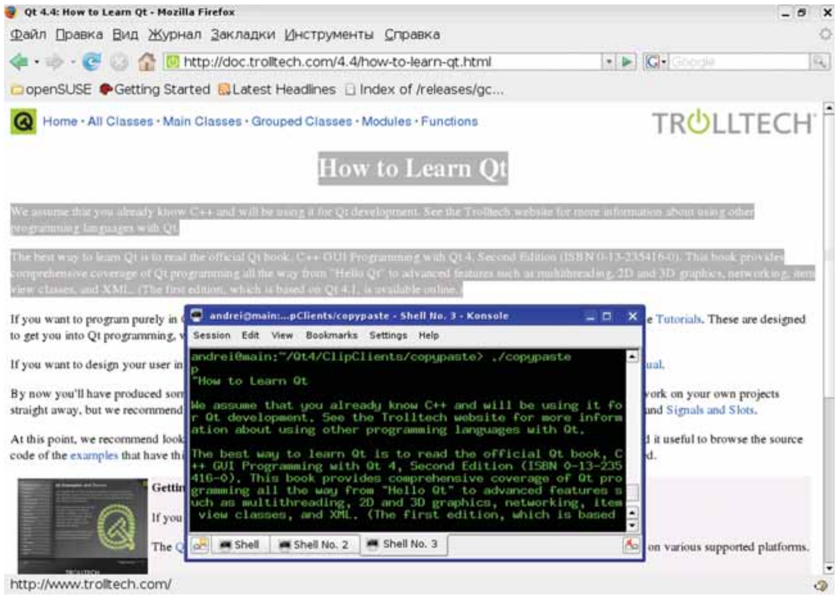
Использование класса QDBusInterface

Вы, наверное, заметили, что в отношении работы с D-Bus программа *sendlines* не проще, чем написанная на языке C. Пришло время показать, как *Qt* действительно может упростить работу с D-Bus. В идеале нам просто хотелось бы иметь класс с тем же набором методов, что и у интерфейса удаленного приложения, так, чтобы при вызове метода локального класса вызывался соответствующий метод интерфейса удаленного приложения. В некоторых случаях этот идеал достижим, причем без особых усилий. Это возможно, если у нас есть описание интерфейса на языке XML. Сейчас же мы рассмотрим более общий вариант решения:

```
#include <QtDebug>
#include <QtCore>
#include <QtDBus>
int main(int argc, char **argv)
{
        QCoreApplication app(argc, argv);
        QDBusInterface clipboard("DBus.Manager.QClipboard", "/
QClipboard", "DBus.Manager.QClipboard");
        int ch = 0;
        while ((ch = getchar()) != 'q')    {
                if (ch == 'c')
                        clipboard.call("setContent", "Data sent
from console");
                if (ch == 'p') {
                        QDBusReply<QString> reply =
clipboard.call("content");
                        if (reply.isValid())
                                qDebug() << reply.value();
                        else
                                qDebug() << "Error calling
content()";
                }
                QCoreApplication::processEvents();
        }
}
```

Программа *copypaste*, текст которой приводится выше, представляет собой консольное приложение *Qt*, которое может вставлять строки в буфер обмена и считывать их из него. Команда "c" заставляет программу скопировать в буфер заданную строку текста, а команда "p" читает содержимое буфера (**рис. 3**).

Программа *copypaste* взаимодействует с D-Bus с помощью объекта класса QDBusInterface, который не делает ничего «волшебного» – по сути, его методы просто объединяют некоторые рутинные операции, которые нам приходится выполнять, когда мы имеем дело непосредственно с классами QDBusConnection и QDBusMessage, однако удобство работы значительно возрастает. Самым часто используемым методом класса QDBusInterface является call(), вызывающий метод интерфейса удаленного приложения. В первом параметре call() передается строка с именем вызываемого удаленного метода. Далее следуют 8 параметров типа «ссылка на QVariant» со значениями, присвоенными по умолчанию; они используются для передачи аргументов.



› Рис. 3. Меньше строк – больше возможностей!

Такое решение может показаться не очень элегантным, но оно заметно упрощает работу с удаленными методами. Если у метода удаленного интерфейса меньше 8 аргументов (а таковых, разумеется, большинство), мы избавлены от необходимости конструирования списка аргументов на основе QList. Например, в строке

```
clipboard.call("setContent", "Data sent from console");
```

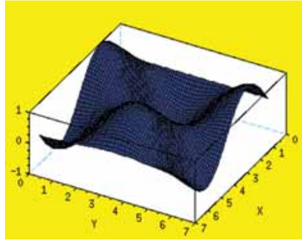
мы вызываем метод setContent с аргументом "Data sent from console". Если же вызываемый метод имеет более восьми аргументов, к нашим услугам – QDBusInterface::callWithArgumentList(), принимающий аргументы вызываемого удаленного метода в виде списка.

Метод call() также используется для вызова удаленных методов, возвращающих значения. При этом приложению-клиенту посылается сообщение определенного типа, которое, помимо прочего, содержит результат вызова. Ситуация осложняется тем, что при вызове удаленного метода может возникнуть ошибка. В этом случае клиенту посылается сообщение другого типа. Для обработки сообщений, возвращаемых удаленными методами, используется шаблон QDBusReply. Его параметром является тип значения, возвращаемого методом. Метод content(), который позволяет прочитать содержимое буфера обмена, возвращает значение типа QString, поэтому объект reply, инкапсулирующий результат вызова метода, имеет тип QDBusReply<QString>. Проверить, не возникло ли в процессе вызова метода ошибки, можно с помощью метода isValid() объекта reply.

В заключение рассмотрим еще одну интересную возможность модуля QtDBus. В статье, посвященной D-Bus, я писал о сообщениях-сигналах и сравнивал их с сигналами *Qt*. Сравнение было более чем уместно, так как в QtDBus действительно существует возможность связывать слот объекта локального приложения с сигналом удаленного приложения. Напомню, что эти сигналы могут быть широковещательными, то есть адресованными не только нашей программе.

Для связывания удаленного сигнала с текущим слотом можно воспользоваться методом connect() класса QDBusConnection. Первые три аргумента метода – соответственно имя сервиса, путь к объекту и имя интерфейса. Далее нужно указать имя сигнала, передать указатель на объект-приемник и строку с именем слота, который будет вызываться в ответ на сигнал. Для отсоединения слота от удаленного сигнала служит метод disconnect() того же класса QDBusConnection.

На этом мы заканчиваем обзор новшеств *Qt 4.x* и переходим на следующий этаж популярной графической системы Linux – KDE 4.x. LXF



Функции: встроенные и внешние

ЧАСТЬ 2 Какая же математика без функций? **Александр Бикмеев** покажет, что может предложить здесь *SciLab*, на примере задачи о теле, брошенном под углом к горизонту.

На прошлом занятии мы узнали, что такое *SciLab*, поработали с матрицами, решили систему уравнений и построили графики. Теперь настало время копнуть чуть глубже, ведь все эти действия можно выполнить и в *OOo Calc*.

Inline-функции

SciLab имеет большую библиотеку встроенных функций, однако вряд ли разработчики включили в нее ту самую, которую необходимо проанализировать вам: например, описывающую перемещение тела, брошенного под углом к горизонту. Ею мы и займемся на данном уроке. Пусть движение тела по оси Oy описывает изменение высоты над поверхностью земли, а по оси Ox – дальность полета. Соответствующие функции имеют вид:

$h(t) = h_0 + v_0\sin(\alpha)t - gt^2/2$,

$s(t) = s_0 + v_0\cos(\alpha)t$,

где v_0 – модуль вектора начальной скорости, h_0 и s_0 – начальная высота и смещение от точек отсчета по вертикали и горизонтали, соответственно, α – угол между вектором начальной скорости и горизонтом.

Эти функции довольно просты, поэтому можно, конечно, вводить их в виде отдельных выражений всякий раз, когда это потребуется. Мы, однако, пойдем другим путем и запишем их в виде так называемых inline-функций, т.е. функций, описываемых в одну команду.

Пусть тело сброшено с высоты 2 км. Вот как можно узнать, на какой высоте оно окажется спустя 5, 10, 15 и 20 секунд.

```
-->deff('[y]=h(t)','y=h0+v0*sin(alpha)*t-g*t^2/2');
-->deff('[x]=s(t)','x=s0+v0*cos(alpha)*t');
-->h0=2000;alpha=(-90*%pi/180);v0=0;g=9.81;
-->t=[5,10,15,20];
-->h(t)
 ans =
  1877.375   1509.5   896.375   38.
```

Значение угла (-90^0) необходимо перевести в радианы: *SciLab*, как и многие системы, использует именно их. Если вы твердо хотите работать с градусами, то можете переопределить h(t) и s(t) следующим образом:

```
-->deff('[y]=h(t)','y=h0+v0*sin(alpha*%pi/180)*t-g*t^2/2');
-->deff('[x]=s(t)','x=s0+v0*cos(alpha*%pi/180)*t');
```

Поскольку любая функция – это, практически, переменная, то и вычисляться она будет в соответствии с описанием, заданным последним. Однако мы далее будем работать с радианами, то есть с первой версией функций.

Подобным образом можно описать любые некусочно заданные функции. Общий вид таков:

```
deff('[<результат> = <имя функции>(<входные переменные>)]','<мат. описание функции>','ключ')
```

Ключ – необязательный параметр, принимающий значения c или n. В первом случае функция будет скомпилирована, что повышает быстродействие, а во втором – нет; по умолчанию действует ключ c. Однако не следует считать, что inline-функции годятся только для простых случаев. Математическое описание функции, а также результат, могут представлять собой матрицу и содержать многошаговые вычисления. Возьмем, например, функцию Qroots(), решающую квадратные уравнения:

```
-->deff('[x1, x2]=Qroots(Qa,Qb,Qc)',['D=Qb^2-4*Qa*Qc';'x1=(-Qb+sqrt(D))/(2*Qa)';'x2=(-Qb-sqrt(D))/(2*Qa)']);
```

Опробуем ее на нашем примере. Пусть начальная высота равна нулю, а скорость равна 20 м/с и направлена вверх (90^o). Определим, когда тело окажется на высоте 20 м. Если ось Oy направлена вертикально вверх, то уравнение будет иметь вид:

$$20 - 20t + \frac{9,81t^2}{2} = 0$$

Найдем его корни при помощи Qroots:

```
-->[h1,h2] = Qroots(9.81/2,-20,20)
 h2 =
   1.7577156
 h1 =
   2.3197563
```

Решим уравнение с другими коэффициентами:

```
-->[q1,q2]=Qroots(1,2,3)
q2 =
  - 1. - 1.4142136i
q1 =
  - 1. + 1.4142136i
```

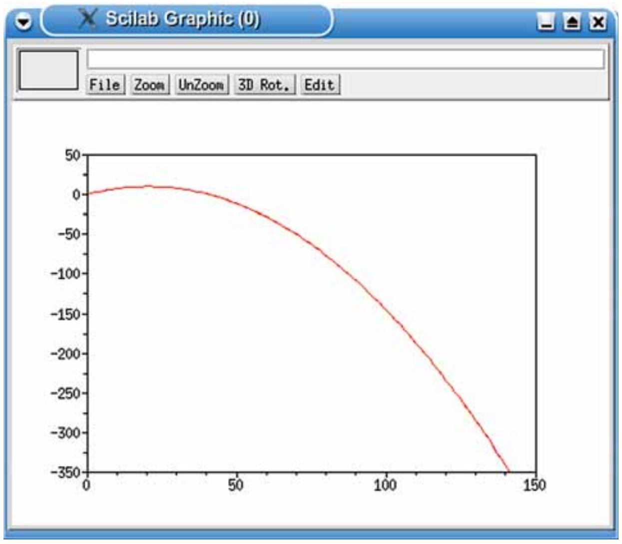
Как видно из этого примера, в тексте описания функции можно создавать внутренние переменные. И кроме того, нам не пришлось обрабатывать случай отрицательного дискриминанта, поскольку *SciLab* понимает мнимые числа и умеет работать с ними. Впрочем, он также умеет решать алгебраические и трансцендентные уравнения и без нашей помощи, так что вернемся к примеру с бросанием.

Давайте построим траекторию полета тела в течение первых 10 секунд при начальной скорости 20 м/с и угле $\alpha = 45^0$, считая,

Месяц назад Мы посмотрели, что такое *SciLab*, и попробовали свои силы в нем.

что $h_0 = s_0 = 0$. Для этого нам необходимо создать вектор значений времени (*t*), а затем построить график, где по оси *Ox* будет отложена *s(t)*, а по оси *Oy* – *h(t)*:

```
-->h0=0;s0=0;alpha=(45*%pi/180);v0=20;g=9.81;
-->t=0:0.1:10;
-->plot(s(t),h(t),'r-')
```



› Рис. 1. Траектория движения тела, брошенного под углом к горизонту и… ушедшего в недра земли.

В результате мы получили траекторию полета тела, которая представляет собой график функции h(s), заданной параметрически. Прочитавшие наш урок в LXF106, вероятно, обратят внимание на добавку r- в вызове plot(). Это специальный параметр, регулирующий оформление. Видите, линия на рис. 1 не черная, а красная? Остальные значения приведены во врезке *Форматируем график*.

Итак, наше тело ушло в недра земли. Чтобы этого не случилось, мы должны грамотно задавать диапазон расчета, т.е. определить, где высота становится равной нулю. Элегантное решение мы разработаем чуть позднее, а пока воспользуемся встроенными средствами.

Функция fsolve позволяет решать линейные и нелинейные уравнения, а также их системы, заданные функциями, то есть находить нули функций. Поскольку ищется численное, а не аналитическое решение, то в результате мы получим одно число. Однако какое решение мы найдем, зависит от значения начального приближения. Так происходит потому, что, как правило, методы численного решения алгебраических уравнений позволяют найти ближайший к указанной точке корень.

Общий вид вызова функции таков:

```
eqAns = fsolve(vInit,F,Fj,e)
```

где eqAns – это переменная или вектор решений, vInit – начальное значение или вектор начальных значений, F – функция или список функций, представляющих собой левые части уравнений, Fj – это производная или список производных функций F, e – точность. Последние два параметра необязательны.

Чтобы корректно построить траекторию, нам необходимо найти нетривиальный положительный корень уравнения h(t) = 0. Зададим начальное приближение, равное 5.

```
-->t_end=fsolve(5,h)
 t_end =
   2.8832081
-->t=0:0.01:t_end;
-->plot(s(t),h(t),'r-')
```

Выполните эти команды самостоятельно и посмотрите результат.

Полиномы

Конечно, искать корни по одному – это неплохо, но как быть, если необходимо узнать их оба сразу? Здесь поможет другой подход, не использующий inline-функций. Если внимательно посмотреть на h(t), то можно заметить, что она представляет собой полином второй степени относительно t. В *SciLab* встроено достаточно инструментов для работы с полиномами – в частности, есть функция root, возвращающая их корни.

Перед использованием полином необходимо определить. Делается это при помощи функции poly. Ее первый аргумент – это вектор-строка коэффициентов или корней полинома (смысл определяется флагом), второй – символьная переменная, а третий – флаг, принимающий значения roots или coeff. Например, пусть нам известно, что начальная высота полета равна 10 м, начальная скорость – 10 м/с и угол равен 30°. Требуется записать закон изменения высоты со временем.

```
-->hCoeff = [10, 10*sin(30*%pi/180),-9.81/2];
-->h=poly(hCoeff,'t','coeff')
 h =
 10 + 5t - 4.905t²
```

Здесь мы указали флаг coeff, сообщающий, что первый параметр – это вектор коэффициентов полинома. А теперь определим, сколько секунд тело уже находится в полете и когда оно упадет на землю.

```
-->roots(h)
 ans =
  - 1.006401
    2.025769
```

То есть тело было брошено примерно секунду назад и упадет примерно через две секунды.

Теперь попробуем пойти другим путем. Известно, что тело находится в полете 3 секунды и через 5 секунд упадет на землю. Определить закон изменения высоты.

```
-->hRoots = [-3,5];
-->h=poly(hRoots,'t','roots')*(-9.81)/2
 h =
    73.575 + 9.81t – 4.905t²
```

Таким образом, в настоящий момент времени тело находится на высоте 73,5 метра и имеет вертикальную скорость, равную 9,81 м/с. В данном случае флаг roots (используется по умолчанию) означает, что первый параметр – это корни полинома, который следует сформировать.

Естественно, работа с полиномами не ограничивается этими двумя функциями. В версии 4.1.2 их насчитывается более 35, среди которых нахождение эрмитовой формы (hermit), полиномиальное деление (pdiv), рациональное упрощение (simp) и многое другое – посмотрите раздел *Polinomial calculations* в справке. Кроме того, с полиномами можно работать так же, как с обычными переменными, то есть вычитать, умножать, возводить в степень и т.д. при помощи символов обычных арифметических действий. Например, определим два полинома, а затем выполним с ними все четыре арифметических действия, а также возведем один из них в квадрат.

```
 -->p1=poly([27,0,0,-8],'s','coeff')
 p1 =
    27 - 8s³
-->p2=poly([9,6,4],'s','coeff')
 p2 =
    9 + 6s + 4s²
-->p_add=p1+p2
 p_add =
    36 + 6s + 4s² - 8s³
-->p_decr=p1-p2
 p_decr =
    18 - 6s - 4s² - 8s³
-->p_div=p1/p2
 p_div =
    3 - 2s
    ------
```

1

-->p_mult=p1*p2

p_mult =

$243 + 162s + 108s^2 - 72s^3 - 48s^4 - 32s^5$

-->p_degr=p2^2

p_degr =

$81 + 108s + 108s^2 + 48s^3 + 16s^4$

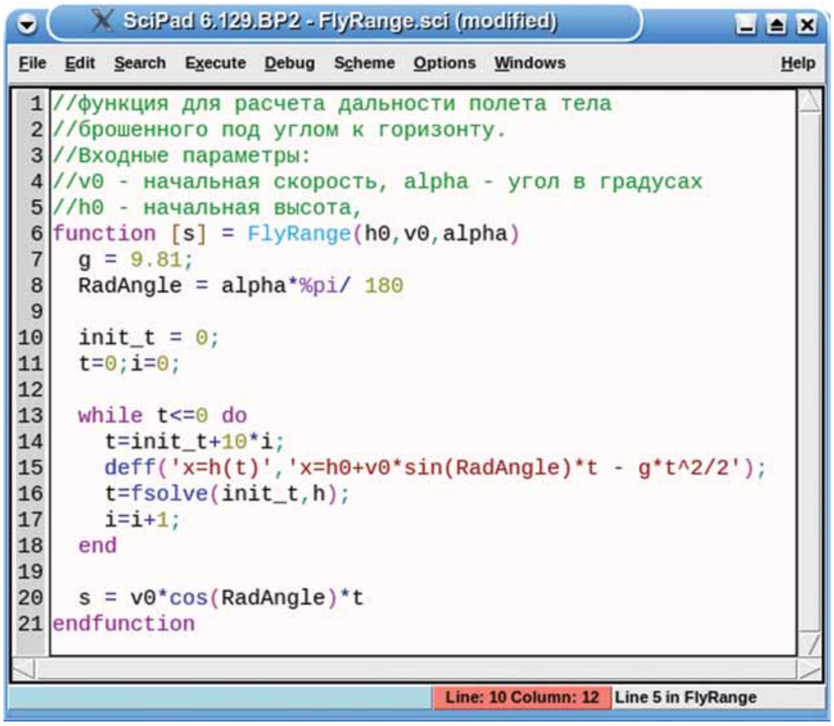
Внешние функции

Да, возможности *SciLab* достаточно обширны, хотя их пока все-таки нельзя сравнивать с функционалом *Matlab*; и все же у этих двух программ есть нечто общее – их часто называют *системами разработки высокотехнологичных приложений*. А это значит, что у *SciLab* (как и *MatLab*) есть встроенный язык программирования. На самом деле, все функции, которые мы использовали, также входят в этот язык, наряду с обычными конструкциями (условие, цикл) и возможностями использования графики. Следовательно, программист, работая в *SciLab*, может не отвлекаться на реализацию численных методов, а использовать уже готовые наработки.

Как и любая среда разработки, *SciLab* обладает текстовым редактором с подсветкой синтаксиса – *SciPad*. Вызывается он выбором пункта меню Editor или командой editor. Редактор, в отличие от оболочки *SciLab*, способен отображать комментарии, набранные кириллицей, однако русского интерфейса не имеет. Вот здесь вы и можете поучаствовать в проекте, поскольку перевод выполняется достаточно просто, а файл не так уж велик. Описание того, как сделать перевод, можно вызвать из *SciPad* (Help > Adding translation...) Желающие – вперед!

Интерфейс предельно прост: строка текстового меню, а под ней рабочая область. Вот в меню-то и сосредоточена вся сила *SciPad*; кроме того, почти все его команды имеют «горячие» клавиши. Пункты меню достаточно стандартны, но хочется отметить, что этот редактор имеет средства отладки (пункт меню Debug), настраиваемую подсветку синтаксиса, проверку парности скобок и прямую выгрузку результата в *SciLab* на выполнение.

При помощи текстового редактора *SciPad* мы сможем создать более элегантное решение для расчета дальности полета тела, брошенного под углом к горизонту (о чем упоминалось ранее). А именно, мы напишем внешнюю функцию, которую можно будет подгружать во время работы и пользоваться ею так же, как и встроенными. Кстати, именно так и выполнены все расширения и многие встроенные функции системы *SciLab*. Чтобы убедиться в этом, откройте каталог, куда был установлен *SciLab*. Если вы используете двоичную версию с **LXFDVD**, то это папка, в которую вы распаковали архив; если же вы устанавливали программу через менеджер пакетов вашей системы, то поищите в **/usr/lib** каталог с именем **scilab-4.1.2**, в моем дистрибутиве (Mandriva 2008.1) он располагается именно там. Затем перейдите в каталог **macros**, и вы увидите множество каталогов, имена которых напоминают названия блоков функций в справке (elem – *Elementary Function*, optim – *Optimisation and simulation* и так далее). Внутри них находятся файлы с расширениями **.bin** и **.sci**: первые – это откомпилированные версии функций, а вторые – исходные тексты. Я рекомендую всем покопаться здесь, поскольку именно так, в отсутствие справочных материалов, можно не только досконально изучить принципы создания приложений в *SciLab*, но и выяснить причины возникновения ошибок при использовании стандартных функций. Может оказаться, что реализованный разработчиками метод (все мы не без греха) неэффективен или содержит ошибку, тогда вы сможете его исправить. Кроме того, настоятельно рекомендую посмотреть содержимое каталога **/usr/lib/scilab-4.1.2/demos**, в котором находятся исходные тексты всех демонстрационных примеров.



› Рис. 2. Встроенный редактор *SciPad* с текстом нашей функции.

Вернемся к нашей задаче. Полный текст функции приведен на рис. 2. Первые строки, начинающиеся с двойного слэша (//) – это комментарии. Затем идет заголовок функции:

```
function [<список выходных параметров>] = <Имя Функции>(<список входных параметров>)
  <тело функции>
endfunction
```

В дальнейшем именно *Имя функции* и будет использоваться в окне *SciLab*. Все остальное является локальным содержимым и на рабочую среду оказывать влияния не будет. Следует отметить, что если при определении функции не указаны выходные параметры, то есть функция ничего не возвращает, то она будет являться процедурой. Как правило, в виде процедур оформляются такие участки программы, как отрисовка графика или картинки, ввод и проверка данных и т.д.

В следующих двух строках мы определяем значение ускорения свободного падения и переводим угол из градусов в радианы. Далее, поскольку мы будем использовать функцию fsolve, задается начальное приближение и устанавливается значение времени, равное нулю. Как уже говорилось ранее, fsolve ищет решение, ближайшее к начальному приближению, поэтому его необходимо задать так, чтобы найденное время было больше нуля. Для этого мы увеличиваем начальное приближение (переменная Init_t) на 10 в цикле до тех пор, пока найденное решение (переменная t) не станет положительным. Таким образом мы сможем найти момент, когда тело упадет на землю. А затем, по формуле для равномерного движения, мы вычисляем, какое расстояние пройдет тело за это время вдоль горизонтальной оси.

Результаты вычисления заносятся в переменные, являющиеся выходными параметрами. В нашем случае он один – s. На этом выполнение функции заканчивается, и результат передается на выход. Теперь нам осталось записать функцию в файл. Для этого выберите последовательно пункты меню File > Save as... или воспользуйтесь комбинацией клавиш Ctrl+Shift+S и сохраните файл в своем домашнем каталоге под именем **flyrange.sci**.

В следующий раз, когда нам понадобится эта функция, мы сможем подключить ее при помощи процедуры getf, которая принимает полный путь и название файла функции. Предположим, что вы (firstuser) сохранили файл в своем домашнем каталоге. Тогда команда будет такой:

```
getf /home/firstuser/flyrange.sci
```

После ее выполнения вы сможете использовать вашу собственную функцию в вычислениях. Кроме того, функцию можно загрузить в *SciLab* непосредственно из редактора *SciPad*, для чего необходимо последовательно выбрать пункты меню Execute > Load into Scilab или

нажать комбинацию клавиш Ctrl+Shift+l. После этого весь исходный код будет передан в *SciLab* и выполнен. Если вы создали в редакторе не функцию, а просто последовательность команд, она будет выполнена, а результат – выведен на экран.

Конечно, лучше было бы самостоятельно реализовать численный метод решения уравнения, но выбранный нами подход позволяет получить компактный код, а также демонстрирует возможность использования встроенных функций при разработке программ.

Отладка

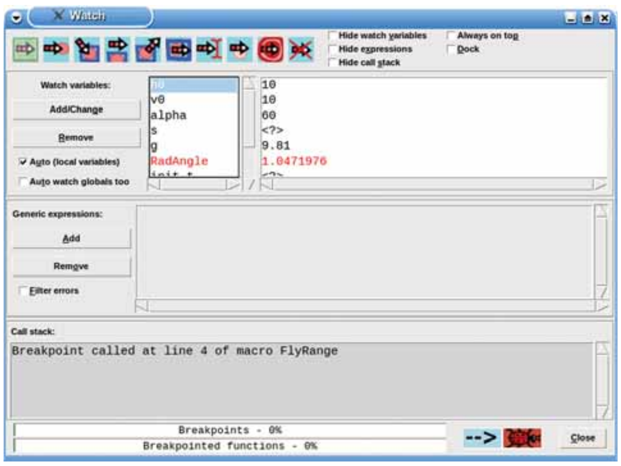
Если при создании вашей собственной функции или процедуры у вас возникли трудности: неверный результат, ошибка при вычислении или что-то еще, то вы всегда можете выполнить программу в режиме пошаговой отладки.



› Рис. 3. Окно для ввода значений входных параметров перед запуском режима отладки.

Прежде чем перейти в режим отладки, вам следует определить значения входных параметров, поскольку в противном случае функция не сможет работать. Для этого либо последовательно выбираем пункты меню Debug > Configure Execution..., либо нажимаем клавишу F10. В данном случае появится окно, показанное на **рис. 3**. В верхней строчке вы можете указать функцию, из которой следует брать входные параметры (если таковая имеется), и нажать кнопку Obtain. Если же входные параметры должны вводиться пользователем, придется указать их вручную. Для этого следует выбрать переменную из списка и нажать кнопку Add/Change. В появившемся окне, в первой строке отображается имя переменной, а во второй – вводится ее значение. Кроме того, если какой-либо переменной нет в списке, вы можете добавить ее, просто изменив имя в первой строке. Нажатие второй кнопки (Remove) приводит к удалению выделенной в данный момент переменной.

Итак, пусть входные параметры определены. Нажмите кнопку OK и закройте это окно. Теперь, если вы вновь выберете пункт меню



› Рис. 4. Окно наблюдения в режиме отладки нашей функции. Установлен флажок автоматической загрузки в окно всех локальных переменных.

Debug, то увидите, что практически все пункты меню стали активны. Вы можете установить/убрать точку останова в текущей строке (F9), удалить все точки останова (Ctrl+F9), выполнить до курсора (Ctrl+F11) и так далее. Пошаговое перемещение выполняется при помощи клавиш F8 (без захода в подпрограммы) Shift+F8 (с заходом в подпрограммы). В ходе выполнения программы вы также можете просматривать значения локальных переменных. Для этого следует вывести на экран окно наблюдения (Watch), нажав Ctrl+F12 или Debug > Show watch. Окно наблюдения содержит три области (**рис. 4**):

- » верхняя – для наблюдения за значениями переменных. При помощи кнопки Add/change вы можете добавлять и изменять переменные. Кроме того, здесь есть две очень полезные опции: Auto (local variables) и Auto watch globals too. Выбор первой позволяет автоматически перенести в окно наблюдения все локальные переменные, а второй – глобальные. Как видно из **рис. 4**, последнее вычисленное значение отображается красным цветом.
- » срединная область для проведения вычислений с текущими значениями переменных.
- » нижняя область для отображения стека вызовов.

Таким образом, вы можете полностью контролировать процесс выполнения.

Конечно же, рамки статьи не позволяют мне описать все, что хотелось бы, особенно это касается программирования. Но я надеюсь, что данная статья даст вам некоторое представление о возможностях и принципах работы *SciLab* с функциями. Если вас заинтересовала эта тема, напишите нам на letters@linuxformat.ru – возможно, мы выпустим несколько уроков по программированию в *SciLab*. LXF

Форматируем график

Опции форматирования линии двумерного графика в *SciLab* разбиты на три категории: цвет линии, тип линии, маркер значения. Они указываются в функции plot в кавычках в любом порядке после массивов с данными для построения. Ниже приведена таблица с описанием каждого из параметров.

Цвет линии		Тип линии		Маркер	
Опция	**Описание**	**Опция**	**Описание**	**Опция**	**Описание**
r	красный	-	Сплошная	+	Знак плюс
g	зеленый	--	Пунктирная	o	Круг
b	синий	:	Точками	*	Звездочка
c	голубой	-.	Пунктир с точкой	.	Точка
m	пурпурный			x	Крест
y	желтый			s	Квадрат
k	черный			d	Ромб
w	белый			^	Треугольник острием вверх
				v	Треугольник острием вниз
				>	Треугольник острием вправо
				<	Треугольник острием влево
				pentagram	Пятиконечная звезда
				none	Нет (по умолчанию)

» Через месяц Мы посмотрим, есть ли конкурент надстройке Simulink пакета Matlab.

Sun Studio Применение встраиваемых шаблонов для повышения производительности программы

Идем на дело

ЧАСТЬ 2 Мы покончили с теорией, и пришла пора взяться за дело – сегодня **Станислав Механошин** покажет, как использовать встраиваемые шаблоны Sun Studio в реальном проекте!

Недавно мне пришлось столкнуться с проектом, в котором изрядную долю времени занимала многократно вызываемая функция *memset16*, которая, как легко догадаться по названию, ведет себя аналогично стандартной функции *memset*, за исключением того, что записывает в память не одинаковые байты, а одинаковые слова. Библиотечная *memset* хорошо известна компилятору и близка к оптимальной, но с *memset16* все оказалось не так легко.

Мне требовалось не только написать оптимальный код для этой функции, но и добиться ее встраивания по месту вызова, чтобы уменьшить накладные расходы на сам вызов, т.к. результаты профилировки показали, что данная функция вызывается часто и обычно с небольшим количеством данных. При подобной постановке задачи выбор определенно падает на встраиваемые шаблоны.

Описывать функцию не надо, т.к. она уже используется приложением. Иными словами, мы имеем дело с прозрачной заменой определения функции:

```
extern uint16_t* memset16(uint16_t* s, int c, size_t len);
```

Итак, создадим ее шаблон:

```
        .inline memset16,0

;/ обнулить старшую часть регистра с символом-заполнителем
        movzwq  %si,%rsi
;/ получить 4 копии символа в rax
        movq    $0x0001000100010001,%rax
        imulq   %rsi,%rax
;/ запомнить указатель на массив для возврата
        movq    %rdi,%r8
;/ количество итераций...
        movq    %rdx,%rcx
;/ по 4 слова за итерацию
        shrq    $2,%rcx
;/ заполнение содержимым rax памяти по адресу rdi
        repnz
        stosq

;/ заполнить остаток, не кратный 4-м словам
        movq    %rdx,%rcx
        andq    $3,%rcx
        repnz
        stosw
;/ возвращаемое значение – адрес массива
        movq    %r8,%rax

        .end
```

В данном случае я использую строковые функции для записи памяти, причем для повышения производительности первая часть функции записывает по 8 байт за раз, и лишь не кратный восьми остаток пишет в память словами.

Обратите внимание, что параметры принимаются соответственно в регистрах rdi, rsi и rdx, а возвращаемое значение помещается в регистр rax.

Теперь требуется перекомпиляция всего приложения с добавлением **memory.il** (имя файла с шаблоном) к, например, CFLAGS в Makefile.

Использование такого шаблона в моем случае позволило улучшить производительность приложения примерно на 15%. Однако при переходе с AMD на платформу Intel выяснилось, что использование записи по 8 байт за инструкцию – не самое выгодное решение. Поэтому для Intel-платформ была создана отдельная версия шаблона, более длинная, но использующая запись по 16 байт за один раз:

```
        .inline memset16,0

;/ запомнить указатель на массив для возврата
        movq    %rdi,%r8
;/ проверка длины на ноль
        testq   %rdx,%rdx
;/ и выход, если ноль
        je      9f

;/ обнулить старшую часть регистра с символом-заполнителем
        movzwq  %si,%rsi
;/ получить 4 копии символа в rsi
        movq    $0x0001000100010001,%rcx
        imulq   %rcx,%rsi
;/ если длина массива меньше 288
        cmpq    $288,%rdx
;/ перейти к заполнению строковыми инструкциями
        jbe     5f

        movq    %rdi,%rcx
;/ проверка адреса массива на выравнивание на 16 байт
        andq    $15,%rcx
;/ переход к заполнению movdqa, если адрес выровнен
        je      2f
        shrq    $1,%rcx
;/ если адрес нечетный, т.е. выравнивания не добиться
;/ перейти к заполнению строковыми инструкциями
        jc      5f

;/ количество символов, которые необходимо записать
;/ до достижения выравнивания
        negq    %rcx
        addq    $8,%rcx
;/ и коррекция счетчика на эту величину
        subq    %rcx,%rdx
;/ символ-заполнитель в rax
        movq    %rsi,%rax
;/ и запись максимум 7 символов до достижения выравнивания
        rep
        stosw

;/ получаем 8 копий символа-заполнителя в xmm0
2:      movdq   %rsi,%xmm0
        punpcklqdq  %xmm0,%xmm0
;/ вычисление целого количества записей по
;/ 8 символов из xmm0, которые не переполнят массив
        movq    %rdx,%rcx
        andq    $0x38,%rcx
;/ и коррекция счетчика для проверки в конце цикла
        subq    %rcx,%rdx
```

Месяц назад Мы выяснили, какая польза может быть от ассемблера в наши дни, погрузившись во встраиваемые шаблоны Sun Studio.

```
;/ коррекция адреса массива с учетом смещения в следующем цикле
        leaq   -128(%rdi,%rcx,2),%rdi
;/ и вычисление адреса перехода внутрь цикла для
;/ получения количества итераций, кратного 8
        shrq   $3,%rcx
        negq   %rcx
        leaq   1f(%rcx,%rcx,4),%rcx
;/ переход внутрь цикла
        jmp    *%rcx

;/ 1-байтная корректировка длины 1-го movdqa для корректности
перехода
        nop
;/ 8 записей по 8 символов за инструкцию
3:      movdqa  %xmm0,(%rdi)
        movdqa  %xmm0,16(%rdi)
        movdqa  %xmm0,32(%rdi)
        movdqa  %xmm0,48(%rdi)
        movdqa  %xmm0,64(%rdi)
        movdqa  %xmm0,80(%rdi)
        movdqa  %xmm0,96(%rdi)
        movdqa  %xmm0,112(%rdi)
;/ корректировка указателя
1:      addq   $0x80,%rdi
;/ и счетчика символов
        subq   $0x40,%rdx
;/ и зацикливание, если он больше или равен нулю
        jae    3b

;/ корректировка счетчика с учетом вычитания до цикла
4:      addq   $0x40,%rdx
;/ выход, если счетчик обнулился
        je     9f

;/ запись остатка, не кратного 8 символам или не выровненного
5:      movq   %rsi,%rax
;/ количество итераций
        movq   %rdx,%rcx
;/ деленное на 4, т.к. Запись по 4 слова за раз
        shrq   $2,%rcx
;/ собственно запись в массив по 8 байт
        rep
        stosq
;/ остаток, не кратный 4-м словам
        movq   %rdx,%rcx
        andq   $3,%rcx
;/ запись по одному слову
        rep
        stosw
;/ адрес массива – возвращаемое значение функции
9:      movq   %r8,%rax

end
```

Здесь следует обратить внимание на использование меток. Хотя в шаблоне и можно использовать обычные текстовые метки, этого лучше не делать. Дело в том, что текст шаблона вставляется в получившийся ассемблер компилируемого модуля практически дословно. Если вы используете шаблон только один раз в единице трансляции, ничего страшного не случится; однако если вы вызовете такую функцию хотя бы дважды, то получите сообщение о повторно определенных символах. Здесь на помощь приходят локальные метки, для которых каждый раз создается новый уникальный символ.

Имена локальных меток – это цифры от 1 до 9, причем для адресации такой метки необходимо указать суффикс b или f, что означает ссылку на ближайшую такую метку соответственно назад или вперед, как в примере выше. Таким образом, хотя возможных имен локальных меток всего девять, их можно определить практически неограниченное количество.

Небольшое замечание по использованию имен функций: имя, указанное в шаблоне, должно быть таким, как того ожидает компилятор, т.е. со всеми декорациями, присущими языку.

Оптимизация шаблонов

При использовании высоких уровней оптимизации, текст шаблона совершенно не обязательно будет вставлен в программу «как есть». Ассемблер шаблона так же участвует в платформенно-зависимых оптимизациях, осуществляемых кодогенератором, как и остальное тело функции. В частности, регистры, через которые передаются параметры и возвращается вычисляемое значение, будут выбраны исходя из того факта, чтобы они уже содержали нужные переменные к моменту использования шаблона. Например, рассмотрим простой шаблон, складывающий два числа:

```
.inline add_int,0
addl   %esi,%edi
movl   %edi,%eax
.end
```

и тестовую функцию, складывающую три числа:

```
extern int add_int(int,int);
int sum(int x, int y, int z)
{
   return x+add_int(y,z);
}
```

После компиляции с использованием команды *cc -fast -m64 -S add.c add.il* мы можем увидеть такой ассемблер:

```
sum:
.CG1:  subq      $8,%rsp
/ ASM INLINE BEGIN:    add_int
       leal      (%rsi,%rdx),%eax          ;/ line : 5
/ ASM INLINE END
       addl      %edi,%eax                 ;/ line : 5
       addq      $8,%rsp                   ;/ line : 5
       ret                                 ;/ line : 5
```

Обратите внимание, что параметры y и z функции sum расположены, соответственно, в регистрах rsi и rdx, в то время как у функции add_int они ожидаются в регистрах rdi и rsi. Однако мы можем видеть, что пересылок между регистрами в данном случае не происходит. Более того, не оптимально описанные в шаблоне сложение и пересылка были превращены компилятором в одну инструкцию lea.

К сожалению, из-за особенностей следования фаз оптимизации операции со стеком не были удалены компилятором, как это было бы сделано, используй мы просто сложение. Это показывает, что использование встроенных шаблонов все же накладывает некоторые ограничения на оптимизацию функций, где они используются, хотя эти ограничения и сведены к минимуму.

Существует, однако, случай, когда шаблон будет использован в месте вызова дословно, без изменений и оптимизаций. Это происходит, если транслятор ассемблера шаблона не понимает написанного в нем. В этом случае его текст будет вставлен в результирующий ассемблер как есть, а параметры переданы строго в соответствии с ABI. Так, например, все псевдооперации ассемблера не понимаются транслятором шаблонов. Например, вы можете захотеть вставить выравнивание перед циклом:

```
        .align 16
1:
;/      ...
        decq       %rdi
        jne        1b
```

Это допустимо и может использоваться, однако .align не понимается транслятором, т.к. эта директива известна лишь собственно программе-ассемблеру. В результате такой шаблон будет использован без оптимизации (что не означает, что вся функция, куда он будет встроен, не будет оптимизирована).

Что дальше?

На этом наш экскурс во встраиваемые шаблоны подходит к концу, однако возможности Sun Studio им, разумеется, не исчерпываются. Есть еще что-нибудь, что вы хотели бы узнать (но боялись спросить)? Черкните нам письмо на letters@linuxformat.ru и сообщите об этом!

По следам Стива Джобса



Apple, знаете ли, это не только Mac OS – были и другие интерфейсы, не сумевшие выбиться в лидеры, но и не ставшие аутсайдерами; кое-что из этого до сих пор доступно в Linux. Что именно? Спросите у **Олега Попова**!

Если вы любите программировать свой рабочий стол, например, на языке Lisp, то возможно, вам больше всего подойдет *fvwm2*, который позволяет решать даже самые неожиданные задачи. Я же предпочитаю просто включить компьютер, быстренько его настроить – и вперед, в Сеть, работать. И поэтому очень уважаю *Window Maker* (*WM*).

Сравнение легких оконных менеджеров публиковалось в **LXF103**. Писать обзоры – дело неблагодарное, поэтому я продемонстрирую достоинства *WM* на примере тех простых настроек, которые сложились у меня на протяжении нескольких лет практической работы.

Window Maker – элегантный оконный менеджер, уходящий своими корнями к детищу Стивена Джобса [Steven Jobs] под названием *OpenStep Desktop*. Этот продукт давно уже почил в бозе, но идея оказалась настолько привлекательной, что два бразильских программиста, Альфредо Коджима [Alfredo Kojima] и Дан Паску [Dan Pascu] решили реализовать ее под свободной лицензией. На мой субъективный взгляд, *WM* присущи следующие достоинства:

» Простота и элегантность концепции.
» Красивый внешний вид.
» Легкость настройки – она осуществляется стандартным приложением *WM*.
» Удобство манипулирования окнами как клавиатурой, так и мышью.
» Минимальные требования к системным ресурсам и высокая скорость работы.

Я не гуру и не знаю, что нужно с позиции юзабилити прочим пользователям, но меня лично более чем устраивает тот функционал, который есть в *WM*.

Азбука от W до M

Выглядит *Window Maker* очень необычно, особенно на первый взгляд, однако если уяснить для себя несколько простых понятий, то все очень быстро встает на свои места.

Итак, главное меню вызывается в любом месте экрана правой кнопкой мыши. Так же происходит в *Blackbox* и его потомках. Средняя кнопка мыши открывает список окон, который позволяет переключаться между ними. У каждого окна есть свое собственное меню, которое вызывается нажатием правой кнопки мыши на заголовке. Еще два предмета, которые вызывают любопытство новичков – это док и скрепка.

» Док – это то место, в котором обычно находятся иконки для быстрого запуска терминала и утилиты для настройки самого *WM*. Кроме того, здесь размещаются dockapp'ы, или док-приложения – программки, выводящие информацию в небольшом квадратном окне. Таких дополнений к *WM* написано довольно много, среди них есть всякого рода мониторы загрузки процессора, сетевого трафика, регуляторы громкости и т.д. После покупки платы с FM-радиоприемником я с удивлением обнаружил работающий dockapp для управления даже таким экзотическим устройством. Есть правда и прекрасно-бесполезные, и ужасно-бесполезные вещи: скажем, док-приложение, выводящее какую-то жуткую анимацию или горящее пламя.

Многие дистрибутивы Linux по умолчанию устанавливают *wmCalClock* – docapp, который демонстрирует текущую дату и время в виде листа календаря.

» Скрепка. Назначение этого предмета двоякое, но он не имеет ничего общего с известным, но отправленным на заслуженный покой обитателем *Microsoft Office*. С одной стороны, скрепка служит переключателем рабочих столов, коих может быть несколько, с другой – она выполняет роль места, куда можно прицеплять иконки для часто используемых приложений, т.е. панели быстрого запуска.

Вот это тема!

Начнем, как всегда, с главного – с рюшечек. А как же без них? *Window Maker* является мировым рекордсменом по количеству созданных для него тем: на одном только www.themes.freshmeat.net их можно найти несколько тысяч (а точнее – 3063), а ведь есть и другие коллекции. Такое изобилие обусловлено тем, что темы можно создавать визуально, и это очень несложный процесс. Правда, справедливости ради надо сказать, что большинство тем, сделанных любителями, выглядит довольно нелепо, но попадаются и подлинные шедевры. Я коллекционирую темы и иногда дорабатываю лучшие экземпляры, исправляя цветовые ляпы, и привожу синтаксис тем к текущей версии.

Одна из самых старых и очень красивых тем – это High Ground (**рис. 1**). Как и положено предмету старины, она выставлена в Тауэре [Tower], но не Лондонском, а канадском: www.cs.mun.ca/~gstarkes/wmaker/. Обои созданы Джимом Ворнером [Jim Varner]. В High Ground



› **Рис. 1. Window Maker с темой High Ground – обновленная классика.**

очень хорошая цветовая палитра; мне нравится темный фон и высокий контраст в меню и заголовках окон. Я слегка исправил ее: привел к новому формату *WM* и изменил градиенты заголовков окон и меню.

Чтобы установить новую тему, достаточно распаковать архив в **$HOME/GNUstep/Library/WindowMaker/Themes/**; обычно она представляет собой каталог с именем **название.themed**. После этого, находясь в текущем каталоге тем в терминале, можно дать команду

```
setstyle ./"HighGroundv2.themed"
```

Теперь HighGroundv2 будет темой по умолчанию. Очень часто приходится немного подправлять размер шрифтов: обычно это бывает, если автор темы использовал другое разрешение экрана. Здесь на помощь приходит программа *WPrefs*.

Управляй окнами

Сейчас, когда мы облагородили внешний вид, наша следующая наша задача – обеспечить удобное управление окнами с помощью клавиатуры. Практически все настройки *WM* можно выполнить с помощью программы *WPrefs*: **/usr/GNUstep/Applications/WPrefs.app/WPrefs**, которая является частью *WM*. Для ее запуска можно воспользоваться терминалом или дважды щелкнуть мышью по иконке с отверткой в доке.

Клавиши, доступные по умолчанию, перечислены в **таблице 1**. Я обычно добавляю несколько своих настроек (см. **табл. 2**), которые работают как триггеры: при первом нажатии – окно распахивается, а при втором – возвращается к предыдущему положению. После некоторого опыта у вас выработается привычка манипулировать окнами без мыши. Поверьте, иногда это очень удобно – например, при активном использовании редакторов *gvim* или *emacs*.

Таблица 1. Горячие клавиши *WM*, определенные по умолчанию

Клавиши	Действие
F12	Открыть меню приложений
F11	Открыть список окон
Control+Escape	Открыть меню для окна
Mod1+H	Скрыть активное приложение[1]
Mod1+M	Свернуть активное окно
Mod1+Up	Активное окно наверх
Mod1+Down	Активное окно вниз
Mod1+Tab	Следующее окно из списка
Mod1+Shift+Tab	Предыдущее окно из списка
Mod1+Control+Right	Следующее рабочее пространство
Mod1+Control+Left	Предыдущее рабочее пространство

[1] Некоторые приложения (например, GIMP) используют несколько окон. Все окна одного приложения можно свернуть одной командой.

Примечание. Под кнопкой Mod обычно подразумевается клавиша Alt на стандартной клавиатуре PC.

Таблица 2. Рекомендуемые дополнительные настройки

Клавиши	Действие
Control+Up	Втянуть активное окно[1]
Control+Left	Распахнуть активное окно по горизонтали
Control+Right	Распахнуть активное окно по вертикали
Control+Down	Распахнуть активное окно
Control+F11	Изменить размер окна

[1] Окно сворачивается вверх, и виден только заголовок.

Подгоняем терминал

WM предлагает некоторые средства для управления поведением окон с помощью оконного меню, которое можно вызвать, если нажать правую кнопку мыши, когда курсор находится над заголовком окна, и выбрать

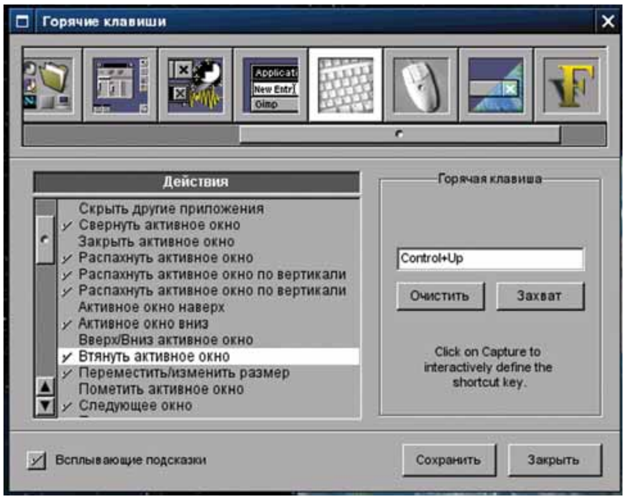


Рис. 2. *WPrefs* позволит настроить все аспекты вашего оконного менеджера.

пункт Свойства.

Очень удобно и эстетически приятно использовать полупрозрачный терминал без обрамления и заголовка окна, который всегда находится внизу и не закрывает запущенные приложения. Так как работа с терминалом для меня практически неизбежна, то я всегда держу окно терминала на одном из рабочих столов.

Обычно я устанавливаю в *WM* для терминала три свойства окна:

- Отключить заголовок
- Убрать рамку
- Удерживать окно снизу

Прозрачность регулируется настройкой самого терминала.

Оборудуем скрепку

Как вы уже знаете, к скрепке можно прицепить иконки для быстрого запуска приложений. Скрепки выдаются по одной на каждый рабочий стол, что наводит на мысль оборудовать последние по целевому назначению. Например:

- Рабочий стол № 1 – Программирование
- № 2 – Сеть и Интернет
- № 3 – Графика

«Отбуксировать» иконку приложения к скрепке можно разными способами, которые будут работать в зависимости от параметров ее настройки, но есть один беспроигрышный вариант:

1. Запускаем приложение.
2. Нажатием правой кнопки мыши над изображением скрепки вызываем меню и выбираем пункт Притягивать иконки, после чего иконка запущенного приложения притянется к скрепке.
3. Чтобы после закрытия приложения его иконка осталась в скрепке, нужно с помощью нажатия правой кнопки мыши над иконкой приложения снова вызвать меню и выбрать пункт Удерживать иконку.

И напоследок

Оригинальный, элегантный, удобный – всеми этими эпитетами можно наградить и *WM*. Но он еще и быстрый и легкий. Даже на очень старых компьютерах *WM* стартует мгновенно, и при этом потребляет столь малое количество памяти, что на современных ПК им можно просто пренебречь. Если поинтересоваться процессами, то можно обнаружить, что *Window Maker* использует их две штуки. Узнать количество занимаемой ими памяти можно с помощью команды **ps**.

На моей рабочей станции с OpenSUSE 10.3 она выдала следующий результат:

```
> ps -w -F   -p 3092,3176
  PID  PPID  C    SZ   RSS PSR STIME TTY          TIME CMD
 3092  3022  0  1691  1340   1 12:38 ?        00:00:00 wmaker
 3176  3092  0  2873  6568   1 12:38 ?        00:00:02 wmaker --for-real=
```

Таким образом, *Window Maker* потребляет меньше 5 МБ основной памяти (4564 КБ, если быть точным). Куда уж тут монстрам-тяжеловесам *KDE* и *GNOME*... **LXF**

Полезные ссылки

- http://www.windowmaker.info/ Домашняя страница *Window Maker*.
- http://dockapps.org Коллекция всевозможных док-приложений.
- www.themes.freshmeat.net Темы для разных оконных менеджеров.

В гостях хорошо,

ЧАСТЬ 3 В мире существуют языки, отличные от английского, и компьютеры попроще IBM Roadrunner. **Никита Шультайс** покажет, как Django справляется и с тем, и с другим.



Наш эксперт

Никита Шультайс
Узнал о Django два года назад на LUG'е и с тех пор использует его как базовый инструмент для разработки web-приложений.

До сих пор мы созерцали «админку» на английском языке, однако ее русификация – тоже не проблема. Для этого нужно всего лишь добавить в кортеж MIDDLEWARE_CLASSES в файле **settings.py** следующую строку:

```
'django.middleware.locale.LocaleMiddleware',
```

Она вызывает специальный обработчик, который отыскивает нужный перевод. Чтобы определить, какой язык требуется, Django делает следующее:

1. Ищет в текущей сессии пользователя значение ключа django_language.
2. Если поиск завершился провалом, Django изучает HTTP-заголовок Accept-Language, генерируемый вашим браузером в соответствии с локальными настройками.
3. Наконец, если ничего не было найдено, используется значение переменной LANGUAGE_CODE, установленное в файле **settings.py**.

Если основной аудиторией вашего сайта будут русскоязычные пользователи, имеет смысл присвоить переменной LANGUAGE_CODE в файле **settings.py** значение ru.

Основным понятием, которым оперирует Django при интернационализации, являются строки перевода – помеченный особым образом текст в ваших исходных кодах или шаблонах. Собственно процесс перевода реализуется библиотекой *gettext*, которая входит в состав Python. Чтобы ваше приложение стало доступным на нескольких языках, нужно пройти несколько этапов:

1. Определить строки перевода в исходных текстах и шаблонах.
2. Запустить утилиту **make-messages.py**, которая найдет все строки перевода и создаст из них языковой файл.
3. Перевести полученный языковой файл.
4. Запустить утилиту **mcompile-messages.py**, которая скомпилирует языковой файл в формат, пригодный для его дальнейшего использования системой.

При этом предполагается, что строки перевода используют английский язык.

Вниманию полиглотов

Сессии в Django полностью основаны на cookie, и если они отключены на клиенте, то могут возникнуть проблемы с переключением языков. Решением является Accept-Language: когда браузер устанавливает этот HTTP-заголовок, он руководствуется выбранными вами языковыми предпочтениями. Например, в *Mozilla Firefox* они находятся в Правка > Настройки > Дополнительно, на вкладке Общее. В разделе Языки, нажмите кнопку Выбрать и переместите Английский язык наверх (см. **рис. 1**). Теперь *Firefox* будет считать, что английский вам роднее, и сообщит об этом Django посредством Accept-Language. Не забудьте вернуть настройки обратно, дабы не пугаться англоязычных интерфейсов ваших любимых сайтов!

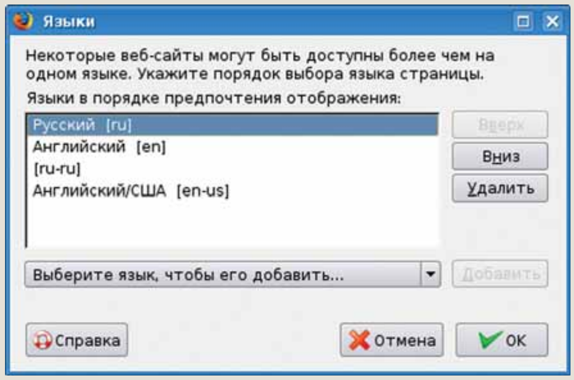


Рис.1. Укажите язык «админки» Django (и не только) в настройках *Firefox*.

Что переводить?

Для начала давайте разберемся, как помечаются строки перевода в исходных текстах приложения. Откройте файл **news/views.py**. В первую очередь, нужно импортировать функцию ugettext() (она, в отличии от gettext(), отлично справляется с Unicode):

```
from django.utils.translation import ugettext as _
```

Как можно заметить, для импортированной функции мы создали псевдоним: _ (подчеркивание). Это было сделано для повышения удобочитаемости кода, содержащего большое число строк перевода.

Месяц назад Мы реализовали возможность комментировать новости и все, что с этим связано (в частности, авторизацию пользователей)

а дома лучше

Затем, давайте передадим функции _() текст, который мы собираемся перевести, и присвоим результат переменной application_name:

```
application_name = _("news")
```

Поместим эту строку в представление last_news, а переменную application_name передадим в шаблон:

```
context = RequestContext(request, {
  "last_news":news,
  "application_name":application_name
})
```

Теперь откройте шаблон **news/templates/news/lats_news.html** и добавьте имя нашего приложения в строку 4:

```
1-3 ...
4   <h2>{{ application_name }}</h2>
5-...
```

В шаблон **news/templates/news/news_detail.html** нужно импортировать тэги интернационализации. Для этого в строке 2 напишите:

```
{% load i18n %}
```

после чего в шаблоне будет доступен тэг {% trans %} (и не только), с помощью которого мы будем отмечать строки перевода. Замените текст в строке 5:

```
5   <h1>{{ news.title }}</h1>
```

на

```
<h1>{% trans "News" %}<br/>{{ news.title }}</h1>
```

Думаю, вы уже догадались, как все это работает. В соответствии с нашим планом, настало время запустить специальную утилиту и создать языковый файл.

Вавилонское смешение

Для начала, откройте терминал, перейдите в директорию приложения news и создайте в ней каталог **locale**:

```
cd news
mkdir locale
```

Утилита **make-messages.py** находится в поддиректории **bin** установки Django. Вызовите ее со следующими параметрами:

```
python /путь/до/django/bin/make-messages.py -l ru
```

Если все пройдет удачно, в директории **news/locale** будут созданы дополнительные каталоги **ru/LC_MESSAGES**, а в них – языковый файл **django.po**, который и нужно переводить. Откройте его в любом текстовом редакторе (для удобства можно пользоваться программой *KBabel* из KDE). Вы увидите следующее:

```
#: views.py:23
msgid "news"
msgstr ""

#: templates/news/news_detail.html:5
msgid "News"
msgstr ""
```

Первая запись в каждом блоке указывает место, где была найдена строка перевода, далее идет текст, который нужно перевести, и наконец, сам перевод (у нас его пока нет). Мы специально создали два варианта слова «news» – с заглавной буквы и с маленькой, чтобы наглядно продемонстрировать, что система воспринимает их как разные слова. Заполните пустые строки русскими «новости» и «Новости». После того, как с переводом языкового файла будет покончено, скомпилируйте его. Для этого воспользуйтесь утилитой **compile-messages.py**, которая опять же входит в состав Django:

```
python /путь/до/django/bin/compile-messages.py
```

В каталоге **LC_MESSAGES** появляется скомпилированный языковой файл **django.mo**. Все готово: наше приложение теперь поддерживает два языка.

Смена языка на сайте

Мы уже выяснили, как задать язык для клиентов с отключенными cookie, однако далеко не все пользователи догадаются так поступить, да и у большинства ваших посетителей cookie так или иначе включены. Значит, чтобы сменить язык сайта, нужно только переписать cookie, которые за это отвечают. И опять Django не оставляет нас в беде. Для начала откроем главный файл с URL-картами – **urls.py**, и после строки

```
(r'^news/',  include('news.urls')),
```

добавим

```
(r'^i18n/', include('django.conf.urls.i18n')),
```

Далее, откроем файл **settings.py** и добавим следующие строки:

```
TEMPLATE_CONTEXT_PROCESSORS = (
  "django.core.context_processors.auth",
  "django.core.context_processors.i18n",
  "django.core.context_processors.request",
)
```

Так мы подключим к нашей системе три контекстных процессора, которые добавляют в шаблоны глобальные переменные для работы с текущим пользователем, переменные языковых настроек и содержимое HTTP-запроса, соответственно. Откройте файл **media/templates/index.html** и после тэга <body> добавьте:

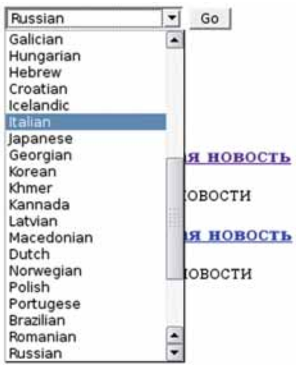
```
<form action="/i18n/setlang/" method="post">
  <input name="next" type="hidden" value="{{ request.PATH_INFO }}" />
    <select name="language">
    {% for lang in LANGUAGES %}
      <option value="{{ lang.0 }}"
      {% ifequal lang.0 LANGUAGE_CODE %}
        selected="selected"
      {% endifequal%}>{{ lang.1 }}</option>
    {% endfor %}
```

```
10        </select>
11     <input type="submit" value="Go" />
```

В строке 1 задается форма, данные из которой будут направлены на встроенное в Django приложение *django.views.i18n.set_language*, расположенное по адресу **/i18n/setlang/** и отвечающее за смену языка сайта. Оно принимает POST-запрос, содержащий следующие переменные:

» **next** – адрес, куда будет перенаправлен пользователь после смены языка. Это, как правило, адрес страницы, на которой мы решили поменять язык. Он содержится в переменной request.PATH_INFO, которую мы передаем в шаблон с помощью контекстного процессора *django.core.context_processors.request*.

» **language** – язык, который был выбран.



› Рис. 2. Многообразие языков, поддерживаемых Django.

Список языков доступен в переменной LANGUAGES, но если вы посмотрите на сайт (**рис. 2**), то увидите огромный перечень, из которого нам нужны только два языка. Дело в том, что если в **settings.py** не определена какая-либо переменная, то ее значение автоматически берется из файла настроек Django – **django/conf/global_settings.py**, а там представлен кортеж LANGUAGES, содержащий все языки, поддерживаемые системой: это более 40 наименований (в том числе, языки, в которых символы пишутся справа налево). Чтобы оставить только те из них, которые нам действительно необходимы, добавим в файл **settings.py** следующий кортеж:

```
LANGUAGES = (
  ('en', gettext_noop('English')),
  ('ru', gettext_noop('Russian')),
)
```

не забыв написать перед ним:

```
gettext_noop = lambda s: s
```

Теперь, если вы загрузите сайт, вам будут доступны только два языка. Вернемся опять к нашей форме: в строке 6 вы найдете новый тэг шаблона ifequal – он сравнивает две переданные переменные, и если они одинаковы, выполняет код в строке 7.

Особенности перевода админки

В отличие от представлений, где мы вызывали функцию ugettext, в «админке» желательно использовать функцию ленивого (отложенного) перевода ugettext_lazy. Ее главное отличие в том, что перевод осуществляется не сразу, а в момент использования строки, например, при обработке шаблона в «админке» Django. Импортируйте эту функцию в файле **news/models.py**:

```
from django.utils.translation import ugettext_lazy as _
```

Каждое поле модели принимает первым позиционным аргументом свое имя, поэтому строку

```
title = models.CharField(max_length=70)
```

можно заменить на

```
title = models.CharField(_('title'),max_length=70)
```

В дополнение к имени поля можно передать текст справки. Замените

```
description = models.CharField(max_length=255)
```

на

```
description = models.CharField(help_text=_('Not more than 255 symbols.'),max_length=255)
```

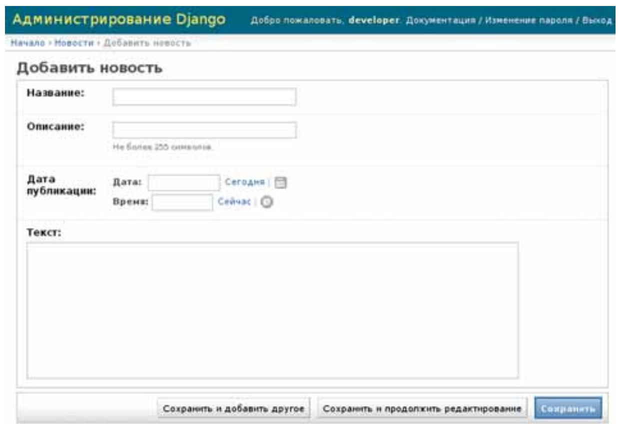
И, наконец, если вам уже надоело слово «News» на главной странице «админки», измените его на более подходящее. Добавьте в класс News подкласс Meta (напомню, что он обеспечивает дополнительные настройки модели), а внутри него объявите переменную, отвечающую за множественное число:

```
class Meta:
   verbose_name_plural = _('news')
```

Добавьте каждому полю имя, текст справки (если требуется) и форму множественного числа. Перейдите в каталог **news** и проделайте те же действия по созданию, переводу и компиляции языковых файлов, что я описывал ранее. После всех операций у вас должно получиться что-то вроде изображенного на **рис. 3 и 4**.



› Рис. 3. Русифицированная главная страница админки...



› Рис. 4. ...и форма добавления новостей.

Кэширование

Наиболее слабыми звеньями работы сайта являются доступ к базе данных и компилирование шаблонов, поэтому стоит сразу задуматься о кэшировании данных. Django прекрасно справляется этой задачей и позволяет создавать и хранить кэш самыми различными способами и в самых разных местах:

» **В оперативной памяти.** Такая возможность достигается с помощью программы *Memcached*, которая служит промежуточным звеном между данными в памяти и Django. Несомненно, этот способ является наибыстрейшим, однако ваш сервер должен обладать достаточным количеством памяти, чтобы хватило и для Django, и для кэша.

» **В базе данных.** В этом случае о скорости доступа к данным нельзя говорить однозначно, так как тут нужно учитывать настройки самой БД; однако если все сделано с умом, можно ожидать хороших результатов.

» **В файловой системе.** Самый простой вариант. Его мы и будем использовать.

Откроем файл **settigns.py** и объявим в нем переменную:

```
CACHE_BACKEND = 'file:///путь/до/кэша'
```

Слово file означает, что мы собираемся использовать файловую систему (против memcached и db для оперативной памяти и СУБД, соответственно), далее идут три слэша и путь до директории, в которой вы собираетесь хранить данные.

Django предоставляет высокоуровневые механизмы для генерации кэша и извлечения его из базы данных. Так, самым верхним объектом является кэш всех страниц сайта. Однако если на странице находятся динамические данные, которые индивидуальны для каждого пользователя, например, строка приветствия («Здравствуйте, Имя»), то все посетители будут видеть приветствие для пользователя, который зашел на сайт самым первым. Если же таких элементов нет, это – самый быстрый способ создать кэш сайта. Чтобы включить его, достаточно добавить строку 'django.middleware.cache.CacheMiddleware' в кортеж MIDDLEWARE_CLASSES в файле настроек. Она должна стоять самой первой в кортеже. На нашем сайте нет приветствия, однако есть возможность смены языков, и после того как кэш страницы создан для одного языка, он будет отображаться и для другого.

Следующим глобальным объектом для кэширования является представление. И опять, здесь мы сталкиваемся с теми же недостатками, что и при кэшировании всех страниц сайта, с единственным отличием: мы сами вправе выбирать, какие страницы будут кэшироваться, а какие – нет. Вы можете импортировать в ваше представление декоратор cache_page:

```
from django.views.decorators.cache import cache_page
```

и применить его к какому-либо представлению, например:

```
@cache_page(60 * 15)
def last_news(request):
  ...
```

Декоратор принимает единственный параметр – время жизни кэша в секундах.

Аналогично можно кэшировать и шаблоны. Откройте файл **news/templates/news/last_news.html** и добавьте после {% extends «index.html» %} строку:

```
{% load cache %}
```

далее, перед {% for news in last_news %} добавьте

```
{% cache 31536000 last_news LANGUAGE_CODE %}
```

а после {% endfor %}:

```
{% endcache %}
```

Сначала мы подключаем дополнительные тэги для кэширования, затем вызываем тэг {% cache %} с тремя параметрами (вообще-то достаточно двух). Первый – время жизни кэша в секундах (у нас это целый год), второй – название, а третий – тоже название, но в нашем случае это переменная, которая вычисляется динамически, а следовательно, для разных языков у нас создается свой кэш. То же самое можно сделать и для приветствия.

Сигналы

Кэширование позволяет сделать сайт очень быстрым, но как быть, если вы определили время жизни кэша в три дня и добавили новость через два? Ведь пока кэш жив, система будет отображать именно его, и нашим посетителям придется ждать целый день. С другой стороны, ставить слишком маленький промежуток тоже нет смысла – тогда от системы кэширования не будет пользы. Предугадать невозможно, да и незачем. Оптимальный вариант – удалять кэш при наступлении какого-либо события (новость добавлена, новость удалена и т.д.). Уведомлением о событии служит сигнал. Создадим внутри директории **news** файл **signals.py** следующего содержания:

```
1  from django.core.cache import cache
2  from settings import LANGUAGES
3
4  def delete_last_news_cache():
5      for lang in LANGUAGES:
6          cache.delete('last_news:'+ lang[0])
```

Обратите внимание на строку 6, где мы генерируем имя кэша. Здесь учтено, что Django соединяет все, что было передано тэгу {% cache %}, символом двоеточия. Поскольку мы не знаем, для каких языков был создан кэш, то удаляем все возможные варианты – в нашем случае их всего два.

Как же сделать, чтобы этот код выполнялся? Проще простого. Откроем файл **news/models.py** и добавим в начало следующие строки:

```
from django.db.models import signals
from django.dispatch import dispatcher
from news.signals import delete_last_news_cache
```

а в конец:

```
dispatcher.connect(delete_last_news_cache,signal=signals.post_save,
sender=News)
dispatcher.connect(delete_last_news_cache,signal=signals.post_delete,
sender=News)
```

Теперь в системе зарегистрированы два сигнала от передатчика (sender) News. Первый сигнал генерируется после сохранения объекта в базе данных, то есть при его создании или изменении, а второй – после удаления. В обоих случаях выполняется функция delete_last_news_cache, определенная выше. Вообще, для обращения к базе данных в Django существует семь сигналов: pre_init, post_init, pre_save, post_save, pre_delete, post_delete и post_syncdb, названия которых говорят сами за себя.

И сверх этого

Кроме возможности создавать кэш ваших данных, Django поддерживает и другие способы оптимизации. Чтобы сделать ваш сайт еще, быстрее нужно добавить в кортеж MIDDLEWARE_CLASSES в файле **settings.py** следующие строки:

```
'django.middleware.http.ConditionalGetMiddleware',
'django.middleware.gzip.GZipMiddleware',
```

Первая из них включает поддержку условных GET-запросов (использование ETag и Last-Modified в HTTP-заголовках), а вторая отвечает за сжатие содержимого для браузеров, которые поддерживают такую возможность (а это все современные реализации).

Уровнем ниже

Низкоуровневое кэширование позволяет хранить любые данные, от переменных и экземпляров классов до фрагментов шаблонов и баз данных. Для работы с ним нужно импортировать в представление модуль cache:

```
from django.core.cache import cache
```

В большинстве случаев вам будет достаточно пользоваться тремя функциями – cache.set для создания кэша, cache.get для извлечения данных, cache.delete для удаления кэша. Рассмотрим их поподробнее:

» cache.set('name', 'value', 60) принимает три параметра: имя кэша, значение и время хранения.

» cache.get('name') – принимает только имя.

» cache.delete('name') – только имя.

» **Через месяц** Мы научимся создавать собственные фильтры для шаблонов, контекстные процессоры и подпрограммы.

Новая серия! Linux в основе домашней киностудии: от захвата с камеры до мастеринга DVD



Kino:

ЧАСТЬ 1 Если друзья отказываются смотреть видеозапись вашего отпуска на Багамах, то это либо из-за черной зависти, либо из-за вашего неумения обработать фильм. С первым ничего не поделаешь, а что касается второго – **Алексей Маслий** к вашим услугам.

Около года назад я приобрел MiniDV-камеру. Естественно, сразу же встал вопрос: как обрабатывать отснятое видео на компьютере и делать из него диски DVD Video? Вариант с простым переписыванием материала с камеры на видеокассеты или DVD я отбросил сразу: поглядев парочку таких фильмов (а они, пожалуй, есть в коллекции у каждого из нас), неизбежно приходишь к выводу, что видео, которое потом предполагается смотреть (а не держать на полке), надо обрабатывать хотя в минимальном объеме.

Разумеется, в комплекте с камерой поставлялся диск с каким-то ПО, предназначенным как раз для этих целей – но, естественно, работающим только под Windows и Mac OS (поэтому он до сих пор и лежит в коробке нераспечатанным). Я же, немного поразмышляв, решил реализовать весь цикл обработки в Linux, тем более, что к тому моменту у меня уже был некоторый опыт (собственно, я знал две вещи: первое – это можно сделать, и второе – какие инструменты для этого требуются). На протяжении следующих нескольких уроков мы поговорим о захвате видео- и аудиопотоков с цифровой камеры, монтаже видеофильма и наложении эффектов (заставки, плавные переходы между сюжетами, титры), кодировании результата в формате MPEG-2 и, наконец, создании DVD-Video с меню, фоновой музыкой и заставками.

Пользуясь случаем, хочу сказать мастерам монтажного стола, что я не занимаюсь профессиональной видеосъемкой, поэтому, если вы ждете увидеть пошаговое HOWTO по созданию собственного блокбастера, рекомендую поискать его в другом месте. Зато для приведения репортажей со скромных и не очень скромных домашних торжеств и офисных праздников в божеский вид этих рецептов вполне хватит. Ну что, приступим?

Подбираем оборудование

Стандартным интерфейсом для DV-камер является IEEE1394, также известный как FireWire. Некоторые модели, включая и мою собственную, оборудуются также портом USB 2.0, но из-за сравнительно низкой пропускной способности данной шины, максимум, что вы сможете получить при таком подключении – это видео VHS-качества, т.е. 320x200 при 30 кадрах в секунду либо 640x480 – при 15. Камера же реально снимает с разрешением стандарта PAL/NTSC (720x576/720x480) при 25 или 30 кадрах в секунду. PCI-адаптер для IEEE1394 в комплекте с кабелем обойдутся вам рублей в 200, так что, даже если на вашем компьютере нет порта FireWire, не стоит экономить на спичках.

Вторым крайне важным показателем является объем винчестера. Поскольку по интерфейсу IEEE1394 передается сырое несжатое видео (то есть каждый кадр в отдельности) плюс звук в формате WAV, то дискового пространства требуется довольно много: 12–14 ГБ на час записи. Дальнейшая обработка и подготовка DVD-диска потребует еще около 10 ГБ (мы вернемся к этому вопросу на последующих уроках). Ну и конечно, диск должен быть настолько быстрым, насколько это возможно для вашего ПК.

Третий фактор – мощность процессора. Вообще говоря, она играет заметную роль лишь на одном этапе, а именно – кодировании в MPEG-2, но разница может быть весьма значительной.

Чтобы не быть голословным, приведу пример. Я попытался смонтировать один и тот же фильм на двух разных компьютерах: AMD Athlon XP 2500+ / 1 ГБ ОЗУ (DDR PC3200 Dual Chanel) / 80-ГБ HDD ATA100 7200rpm и AMD Athlon64 4000+ / 1 ГБ ОЗУ (DDR2 PC5300 Dual Chanel) / 120-ГБ HDD SATA II 7200rpm. Результаты – 4 часа 52 минуты на первой системе и 2 часа 48 минут на второй – говорят сами за себя. Различие становится еще более существенным (не по величине, а по сути), если вы ограничены в машинном времени (например, разделяете домашний компьютер со своими родными).

Производительность дисковой подсистемы особенно критична при захвате видео с камеры. Если скорости только-только хватает на выполнение данной задачи, во время работы к компьютеру лучше не прикасаться. Подумайте также об останове прочих процессов в системе: часть материала может быть загублена, если какому-нибудь *beagled* внезапно придет в голову светлая мысль срочно проиндексировать ваши файлы.

Открытый инструментарий

На данном фронте особого разнообразия не наблюдается (что, вообще говоря, нетипично для Open Source). Безусловным фаворитом является редактор для нелинейного видеомонтажа *Kino* (**www.kinodv.org**), базирующийся, несмотря на заглавную K в имени, на *GTK+*. Кроме захвата видео с камеры, в *Kino* также производится видеомонтаж, наложение эффектов (заставки, титры, переходы) и кодирование видео- и аудиопотоков (с помощью сторонних программ-помощников).

Для создания DVD-дисков (DVD-мастеринга) приложений уже значительно больше. Лично я предпочитаю *QDVDAuthor* (**qdvdauthor.sourceforge.net**) – по сути, оболочку к консольной программе *DVDAuthor*. *QDVDAuthor* позволяет визуально создавать меню (с заставками, анимацией, переходами, фоновой музыкой и т.д.), а затем преобразует все это к формату, необходимому для записи DVD-Video, и генерирует XML-файл с инструкциями для *DVDAuthor*. В качестве альтернативы можно посмотреть *DVDStyler* (**www.dvdstyler.de**), который также является оболочкой к *DVDAuthor* и входит в состав LiveCD **Dyne:bolic** (LXF106).

Для подготовки заставок я буду использовать графический редактор *GIMP* (**www.gimp.org**), для обработки звука (если таковая потребуется) – *Audacity* (**audacity.sourceforge.net**), а для разных вспо-



К захвату готовы!

могательных целей – *Avidemux* (**www.avidemux.org**), видеоредактор, который, помимо прочего, может заменить *Kino* в области кодирования видео/аудио.

На самом деле и *Kino*, и *QDVDAuthor*, и даже *Avidemux* полагаются в своей работе на ряд консольных приложений, таких как *mjpegtools*, *lavc*, *mencoder* и др. За исключением данного случая, я не буду упоминать о них, поскольку внутренняя механика обычно бывает хорошо скрыта от глаз пользователя (по крайней мере до тех пор, пока не вылезет в виде зависимостей и не стукнет больно).

На протяжении всей серии уроков я буду использовать дистрибутив ALT Linux Desktop 4.0.1. Как и всегда, этот факт не будет иметь решающего значения – вы сможете следовать приведенным рекомендациям в любом современном дистрибутиве общего назначения. Разумеется, названия пунктов меню и расположение программ могут слегка отличаться, но общая суть от этого не пострадает.

Приступим к делу

В первую очередь, убедитесь, что с оборудованием все в порядке и не требуется никаких дополнительных действий (это справедливо в подавляющем большинстве случаев, если вы не модифицировали аппаратную часть, т.е. не устанавливали дополнительных устройств и, в особенности, интерфейсных плат IEEE1394 после инсталляции дистрибутива). При необходимости, воспользуйтесь стандартными конфигураторами, чтобы обнаружить и настроить новое оборудование. Для проверки подключите камеру к порту IEEE1394 и включите ее в режиме воспроизведения (Play). Запустите *Kino* (как правило, он обозначен в меню приложений как Мультимедиа > Медиа(Видео) редактор). В меню программы выберите пункт Правка > Настройка и найдите вкладку IEEE1394 в появившемся диалоге. На ней присутствует всего один пункт: Устройство AV/C (правда, в более старых версиях программы число элементов на данной вкладке может быть существенно большим). Если все хорошо, то здесь будет отображаться название вашей камеры (**рис. 1а**), если же нет, то его не будет и/или появится предупреждающее сообщение. Так, на **рис. 1б** изображена ситуация, когда от камеры нет отклика, а на **рис. 1в** – сбой, вызванный отсутствием в системе драйверов для IEEE1394. В первом случае необходимо проверить соединение интерфейсного кабеля с камерой и компьютером и убедиться, что устройство включено и находится в режиме воспроизведения. Во втором необходимо подгрузить соответствующие модули ядра. Эта процедура производится от имени суперпользователя. Откройте терминал и выполните в нем команду:

```
su -
modprobe raw1394
modprobe dv1394
```

Если не было получено сообщений об ошибках, значит, загрузка прошла удачно, но на всякий случай проверьте это, введя команду *lsmod | grep 1394*. Вы получите список загруженных модулей ядра, имеющих отношение к подсистеме FireWire:

```
raw1394          32048  0
dv1394           23312  0
ohci1394         38088  1 dv1394
ieee1394        369784  4 raw1394,dv1394,sbp2,ohci1394
```

Драйвера FireWire входят в стандартную поставку ядра Linux и должны быть доступны в вашем дистрибутиве (возможно, хотя и маловероятно – в виде отдельного пакета). Если же вы собирали ядро вручную, убедитесь, что вы не забыли установить группы параметров CONFIG_FIREWIRE_* и CONFIG_IEEE1394_* в **m**.

Теперь присоедините камеру к компьютеру и включите ее в режиме воспроизведения. Набрав **cat /var/log/messages | grep 1394**, вы увидите сообщения вида:

```
Apr  2 14:34:35 helavista kernel: ohci1394: fw-host0: OHCI-1394 1.1 (PCI): IRQ=[10]  MMIO=[df002000-df0027ff]  Max Packet=[2048]  IR/IT contexts=[4/8]
Apr  2 14:38:05 helavista kernel: ieee1394: raw1394: /dev/raw1394 device initialized
```

Далее, запустите Kino и перейдите в диалог настроек, как было рассказано выше. На вкладке IEEE1394 должно появиться название подключенной камеры, как на **рис. 1а**.

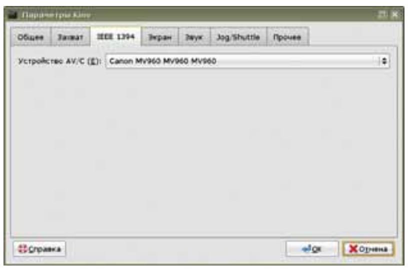
Следует заметить, что большинство современных дистрибутивов подгружают модули ядра при обнаружении нового устройства автоматически, так что в первую очередь попробуйте подключить видеокамеру в режиме воспроизведения и посмотреть настройки Kino.

И последнее – если включить камеру не в режиме воспроизведения, а режиме записи (съемки; иногда он также обозначается как CAMERA), то настройка тоже пройдет удачно, а вот захват работать не будет.

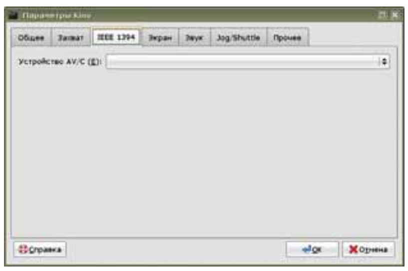
Подготовим рабочее место

Скажу пару слов об интерфейсе *Kino*. Главное окно программы (**рис. 2**) состоит из традиционных элементов: меню, панель инструментов, статусная строка и рабочая область. Последняя разделена на четыре части. В вертикальном столбце слева отображается раскладка видеофрагментов с подробной информацией о них в поряд-

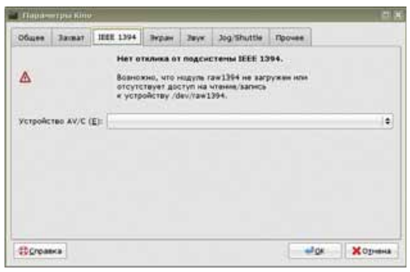




› **Рис. 1а. Все в порядке.**



› **Рис. 1б. Если вы видите это, значит, ваша камера не отвечает...**



› **Рис. 1в. ...а если это – значит, не загружены драйвера IEEE1394.**



ке воспроизведения. В правой части рабочей области вертикально располагаются вкладки рабочих режимов: Редактор, Захват, Шкала времени, Срезка, Эффекты, Экспорт. По умолчанию активирован первый режим, Редактор, а переключаться между ними можно также с помощью основного меню (пункт Вид) или функциональными клавишами F2-F7. Все самое интересное происходит в центральной части, а ее содержимое зависит от выбранного режима работы. В первых четырех случаях здесь находится окно просмотра видео, под которым располагаются стандартные элементы управления (пауза, воспроизведение, перемотка и т.д.), регулятор громкости и временная шкала, видом которой можно управлять из специального выпадающего меню.

Перед тем, как приступить к работе с *Kino*, рекомендую вам внести свои предпочтения в настройки по умолчанию: тогда они будут автоматически действовать на все новые проекты. Конечно, в каждом конкретном случае их можно переопределить, но если вы вдруг забудете это сделать, а такое иногда случается, то домашний каталог захламляться не будет. Диалог настроек, как мы уже знаем, доступен через меню Правка > Настройка. Помимо уже известной нам (и сейчас не особенно нужной) третьей по счету вкладки IEEE1394, там присутствует еще шесть: Общее, Захват, Экран, Звук, Jog/Shuttle, Прочее. Проделайте следующее:

» **Общее** Здесь необходимо выбрать формат видео по умолчанию (PAL или NTSC), частоту дискретизации звука и соотношение сторон (4:3 или 16:9).

» **Захват** Тут следует указать полный путь к файлу, в который будет сохраняться сырой видеопоток. По умолчанию, это **captureXXX.dv** в вашем домашнем каталоге, что, на мой взгляд, неудобно. Далее, выберите тип файла (я оставляю RawDV) и параметры записи. Авторазбивка на файлы означает, что для каждого сюжета (или отрезка записи длиннее 2 ГБ) будет создаваться новый файл.

» **Экран** управляет выводом видеопотока на ваш монитор. Подберите здесь параметры на свой вкус, но на слабых машинах и с видеокартами без аппаратного ускорения я бы рекомендовал активировать Reduced Xvideo и Терять кадры при необходимости и отключить Просмотр во время захвата. Оптимальный для вашего компьютера набор опций составляется путем проб и ошибок.

» **Прочее** Здесь доступны дополнительные настройки программы, из которых нам сейчас будет интересен Каталог проекта по умолчанию (название, полагаю, говорит само за себя).

› **Рис. 2. Главное окно программы *Kino*.**



Теперь, когда наш инструментарий в сборе и настроен как надо, можно приступать к делу.

Захват видеопотока

Запустите *Kino* и выберите вкладку Захват. В этом режиме к стандартному виду над кнопками управления просмотром добавляются еще четыре и строка с именем файла для сохранения (**рис. 3**). Дополнительные кнопки будут активны только в том случае, если камера подключена к компьютеру и работает в режиме воспроизведения.

Проследите, чтобы кнопка AV/C была нажата – это позволит управлять камерой с компьютера. Далее, при помощи кнопок перемотки найдите начало сюжета, который необходимо загрузить. Для навигации по носителю (в случае MiniDV – кассете) удобно пользоваться кнопками Перейти к началу предыдущего/следующего сюжета. Для более точного позиционирования имеются инструменты покадрового сдвига.

Перед началом захвата не забудьте указать имя файла и каталог для сохранения видеоданных. Также желательно выбрать формат индикатора положения в видеопотоке. По умолчанию, показывается порядковый номер кадра, но гораздо удобнее выбрать режим Счетчик, как на **рис. 3**; в этом случае индикатор примет вид часы:минуты:секунды.доли секунды.

Для начала захвата необходимо нажать кнопку Захват (с большой красной точкой). Камера автоматически начнет воспроизведение, которое будет записываться в файлы с указанным названием, с прибавлением номера и расширения **dv**. Если в указанном каталоге уже есть файлы видеоданных *Kino*, нумерация будет начата с первого свободного слота.

Остановить захват можно кнопкой Стоп. Оставшиеся два органа управления позволяют записать стоп-кадр и отключить вывод звука на внешнюю акустическую систему (естественно, аудиопоток продолжает сохраняться).

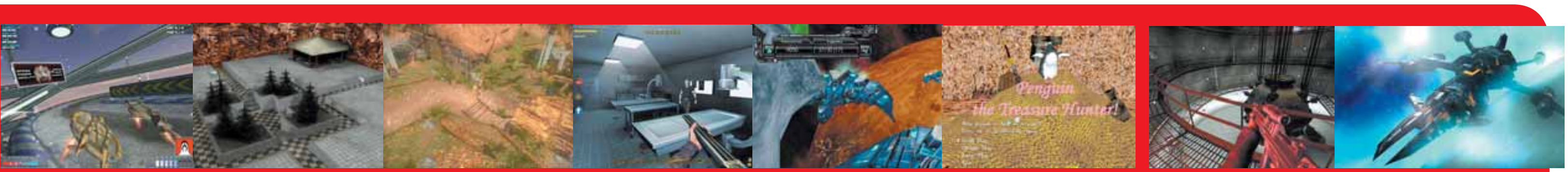
По завершении процесса загрузки материала не забудьте выключить видеокамеру или отожмите кнопку AV/C. В противном случае любые перемещения по загруженному на компьютер видеоряду заставят камеру искать в видеопотоке соответствующее местоположение, что особенно вредно для MiniDV-устройств и их кассетами.

Вот и вся процедура захвата отснятого материала с цифровой видеокамеры – как видите, это несложно. Ну, а про обработку полученного ролика мы поговорим в следующий раз – пока что положите его на полку. LXF



› **Рис. 3. Приступить к захвату!**

» **Через месяц** Монтаж видеоряда средствами *Kino* и наложение различных спецэффектов.



LINUX FORMAT

» Ни строчки кода! Создаем в *Blender* трехмерную игру, не программируя.

LINUX FORMAT На DVD
» «Исходный код» нашей игры.



АНДРЕЙ ПРАХОВ
Участник нескольких игровых проектов, представитель СМИ и начальник www.linuxmedia.ru

Луч света в темном царстве

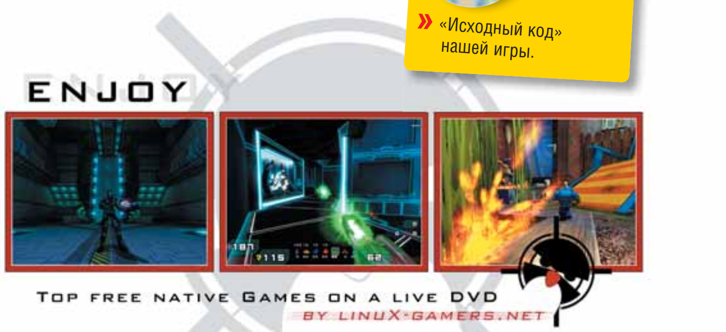
Нельзя сказать, чтобы игровой мир Linux был перенасыщен качественными «хардкорными» играми. Они есть, но по большей степени являются Windows-ориентированными и запускаются в режиме эмуляции. Родные приложения можно пересчитать по пальцам. И хотя имеется немало свободных игровых технологий и под нашу систему, монстры игростроя не торопятся выпускать свою продукцию для Linux. На этом неблагожелательном фоне звездочкой надежды сверкнула новость о серьезной заинтересованности компании Valve Software к портированию своего движка Source на платформу Linux. По сути, разработчикам придется глобально перетряхнуть код движка, так как он изначально был ориентирован на использование *DirectX*.

Source Engine является одним из самых дорогих и технологичных движков, на основе которого выпущены такие игры, как *Counter-Strike: Source*, *Half-Life2*, *Portal*. Вполне вероятно, открывающиеся возможности заинтересуют и команды разработчиков игр.

Среди более доступных технологий можно отметить движок *TGE* компании GarageGames, которой изначально имел порт и под Linux. Вышедшая совсем недавно новая версия шейдерного движка *TGE Advanced*, по слухам, также будет скоро портирована под Linux. Однако стоимость его на пару порядков ниже *Source*.

Подобное положение вещей не может не радовать, но давайте подождем и посмотрим...

info@linuxmedia.ru

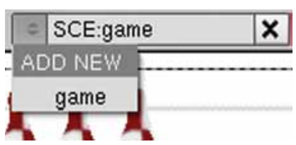


Не желаете ли меню?

ЧАСТЬ 4 Здравствуйте, уважаемые телезрители! В эфире программа «Смак» с Андреем... простите, рубрика «Игрострой» с **Андреем Праховым**. И сегодня мы займемся игровым меню.

Вот мы и вышли на финишную прямую – осталось доделать совсем чуть-чуть, и наша игра будет готова увидеть белый свет. Давайте приступим к созданию игрового меню. Как вы, наверное, помните, в концепт-плане программы главное и единственное меню игры состояло всего из двух элементов: Start и End game. Пунктов, конечно, чертовски мало, и они не отражают всего того, что хотелось бы видеть, однако для понятия самой идеи реализации их вполне хватит.

Прежде чем начать кодирование, необходимо разобраться, как мы планируем выводить меню на экран. Имеется два варианта. Первый – выделить область на игровом пространстве и добавить туда еще одну камеру, а далее, в нужный момент, просто переключаться с игровой камеры на добавочную. Второй вариант – воспользоваться тем, что *Blender* умеет создавать различные сцены и работать с ними. Что это такое? Вкратце можно объяснить так: в пределах одного проекта возможно существование подпроектов с уникальными свойствами, но общими глобальными настройками. Если смотреть на это сквозь призму игростроя, то концепция сцен позволяет реализовать механизм игровых уровней. А чем меню не игровой уровень?



› Рис. 1. Концепция сцен позволяет по максимуму использовать пространство и создавать уровни.

Чтобы создать новую сцену, необходимо выбрать соответствующий пункт из выпадающего меню вверху программы (**рис. 1**). Естественно, там же вы можете переключаться с одной сцены на

» Месяц назад Стремясь добиться большей реалистичности, мы добавили в нашу игру текстуры.

другую. Но запомните, что при генерации двоичного файла игры активная в текущий момент сцена считается главной и запускается первой.

При добавлении сцены вам будет предложено несколько вариантов ее создания:

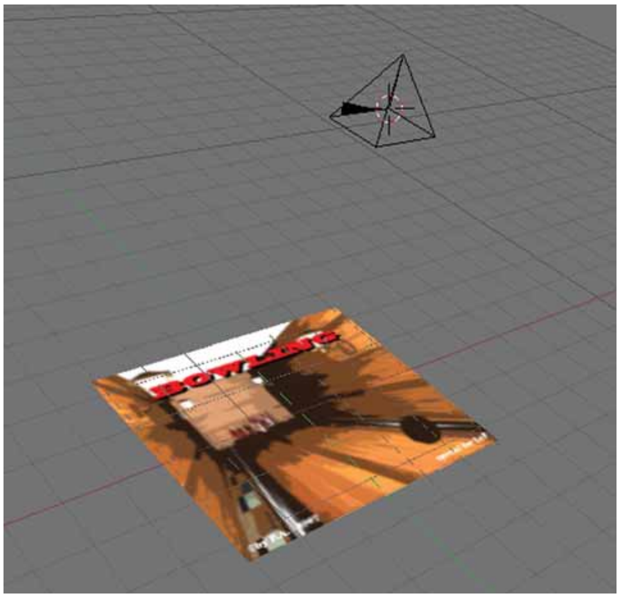
» Empty [Пустая] – равноценно сбросу *Blender* для нового проекта;

» Link Objects [Присоединить объекты] – содержание копируется, однако изменения в одной сцене отражаются и в других, созданных таким способом;

» Link ObData [Присоединить данные объектов] – в отличие от Link Objects, вы можете перемещать объекты, но изменения самих mesh'ей или материалов отразится во всех сценах;

» Full Copy [Полная копия] – создается независимая копия текущей сцены.

В нашем случае оптимальным вариантом является пустая сцена. Для нормального функционирования движка необходимо создать хотя бы одну камеру и один источник света. Как это сделать, описывалось в предыдущих уроках по *Blender*. Поэтому не будем отвлекаться, а займемся созданием основы нашего меню. Итак, добавьте камеру, свет и примитив Plane. Расположите их примерно так, как показано на **рис. 2**. Источник, для лучшего освещения, желательно поместить вблизи от камеры. Впрочем, это несущественно, и вы сами сможете отрегулировать его интенсивность и местоположение. Теперь необходимо расположить примитив так, чтобы он занимал всю активную область, отмеченную в проекции камеры пунктирными линиями. Запустите движок и удостоверьтесь, что цвет примитива закрасил всю видимую область игрового пространства.



› **Рис. 2. Что ж, бывают и лежачие меню.**

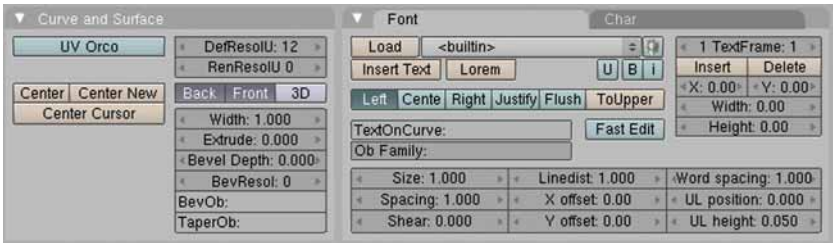
Чтобы наше меню выглядело не столь уж унылым, я создал простой фон (**mainmenu.tga**), который вы сможете найти в каталоге **textures** файлов этого урока, расположенных на **LXFDVD**. Для размещения его на плоскости воспользуйтесь режимом Face Select и редактором UVEditor. Более подробно о работе с текстурами в BGE рассказывалось в LXF105.

Вот теперь заготовка для меню почти готова, но стоп... отсутствует самое главное – текст.

Да будет текст!

Спешу обрадовать любителей трудностей: работа с текстом в *Blender*, а тем более в GE, изобилует непонятными моментами. Да и литературу по этой части отыскать весьма сложно. Опять-таки имеется несколько вариантов. Например, для меню и контента со статичным содержимым можно использовать стандартный способ работы с титрами *Blender*.

Создайте заготовку, выбрав из меню, вызываемого по клавише Пробел, функцию Text. На экране должна появится строка Text, сигнализирующая о запуске встроенного текстового редактора. Одновременно на панели управления откроются инструменты, необходимые для работы (**рис. 3**). Возможности, предлагаемые программой, достаточно широки для комфортной правки текста.



› **Рис. 3. Встроенный текстовый редактор позволяет максимально удобно работать с текстовой заготовкой.**

Вы можете выбрать подходящий шрифт, воспользовавшись клавишей Load, загрузить сторонний текст из файла командой Insert text, форматировать начинку, как вам заблагорассудится, и многое другое. Но запомните следующий момент: все манипуляции с текстом можно проводить только в режиме редактирования. Однако, если вы попробуете запустить игровой движок, то на экране вас встретит кромешная пустота. Дело в том, что *GE* не умеет работать с тем форматом, в котором в данный момент хранится наш текст. Единственный вариант – преобразовать его в обычный mesh. Естественно, в этом случае отредактировать текст будет уже невозможно. Для конвертации выйдите из режима редактирования клавишей Tab и нажмите Alt+C. Из предложенных пунктов меню выберите Mesh.

Полученные титры будут выглядеть гораздо эффективнее, если придать им объем. Обычная функция Extrude позволит достичь задуманного. Войдите в режим редактирования, затем выделите весь объект и нажмите клавишу E. Именно подобным образом сделаны титры в меню демонстрационной версии игры.

Второй вариант использования текста завязан исключительно на игровой движок. В качестве исходного шрифта используется обычный файл **.tga**, содержащий все необходимые изображения букв. Пример такого файла – **font.tga** – вы опять же сможете найти в подкаталоге **textures** файлов этого урока. Подобный подход используют многие игровые движки, так что сложностей с поиском подходящего шрифта у вас возникнуть не должно. Как говорится, Google вам в помощь!

После проведения необходимых манипуляций (см. врезку *Создаем текст в стиле BGE*) необходимо создать текстовую переменную с именем Text (именно так и не иначе!), а в качестве ее содержания впечатать желаемый текст. Естественно, используя логические блоки работы с переменными, вы можете с легкостью, динамически, изменять это содержимое в процессе игры. Давайте рассмотрим это на простом примере.

Итак, предположим, вы создали заготовку по приведенной схеме. По нажатию клавиши Пробел текущее содержание необходимо изменить на фразу Start game. Для этого добавьте к объекту Plane логическую цепочку, состоящую из кирпичиков Keyboard > AND > Property. Как настраивать сенсор клавиатуры, говорилось уже не раз, а сама реализация заключается в подстановке в свойстве Assign названия переменной и необходимого значения. В нашем случае это будут: prop: Text и Value: Start game. Вот и все!

Реализация меню

На данном этапе у вас уже должна быть заготовка, состоящая из новой сцены с текстурированным объектом Plane, текстом и настроенной камерой. Надеюсь, вы уже дали значащие имена имеющимся сценам? Если нет, то сделайте это, так как в дальнейшем нам придется как-то обращаться к ним! В нашем примере они обозначены как сцены game и mainmenu, соответственно.

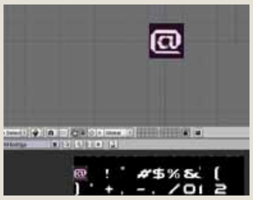
Скорая помощь

После конвертирования текста в mesh вполне могут появиться артефакты, искажающие прорисовку полученного объекта. Это связанно с неправильными расчетами нормалей. Исправить ситуацию можно либо пересчитав нормали командой Mesh > Normals > Recalculate, либо просто перевернув их командой Mesh > Normals > Flip. Естественно, подобные манипуляции должны проводиться в режиме редактирования.

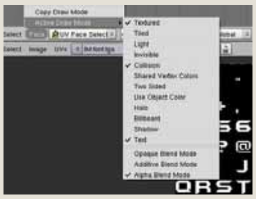
Создаем текст в стиле BGE



Основой для текста служит примитив Plane. Добавьте его стандартным образом: Add > Mesh > Plane.



В режиме Face Select (клавиша F) необходимо расположить первый образ символа «@» так, чтобы он занял всю область примитива. Заметьте, что по умолчанию символ располагается в перевернутом состоянии. Этим можно воспользоваться, чтобы выводить текст, скажем, вертикально. Однако для правильного горизонтального отображения необходимо развернуть область выделения в *Image Editor* так, чтобы символ «@» располагался точно так же, как в файле **font.tga**.



Следующим шагом необходимо указать *GE*, что данный примитив является основой для вывода текста. Войдите в режим Face Select и активируйте пункты Text и Alpha Blend Mode в меню Face > Active Draw Mode. Последний заставляет программу выводить буквы с использованием альфа-канала (т.е. с прозрачным фоном).



Последний штрих – настройка переменной String для базового объекта. Имя переменной должно быть «Text». Таким образом мы указываем программе, что значение данной переменной следует вывести на экран. В поле значения переменной введите необходимое содержание. Теперь смело нажимайте клавишу P для запуска игрового движка и наслаждайтесь результатом!

В качестве указателя мы будем использовать обычную сферу. Расположите ее между основой и камерой так, чтобы она по умолчанию указывала на первый пункт меню.

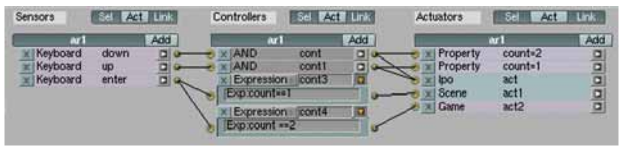
Рассмотрим логику работы. Для перемещения курсора будем использовать однокадровую ключевую анимацию, настроенную таким образом, чтобы в начальном кадре сфера указывала на первый пункт меню, а в следующем – на второй (подробно о ключевой анимации рассказывалось в статьях серии «Blender для начинающих», публиковавшихся в LXF87/88-LXF91). Управлять анимацией можно посредством активатора Ipo, а именно – его подфункции Property. Весь смысл заключается в том, что изменяя значение числовой переменной с 1 до 2, мы указываем программе, какой кадр является активным.

Для более точного понимания работы логики воспользуемся псевдокодом:

```
//Пусть указателю присвоена переменная Integer с названием Count, тогда
Sphera.Count = 1 // активен первый пункт меню
ЕСЛИ нажата клавиша «Down» ТО
        Sphera.Count = 2
        выполнить анимацию
ЕСЛИ нажата клавиша «Up» ТО
        Sphera.Count = 1
        выполнить анимацию
ЕСЛИ нажата клавиша «Enter» AND Sphera.Count ==1 ТО запуск игры
ЕСЛИ нажата клавиша «Enter» AND Sphera.Count ==2 ТО выход из игры
```

Реализация проверки курсорных клавиш и действий по ним не должна представлять для вас особенных сложностей. Для настройки Ipo-активатора необходимо выбрать подфункцию Property и указать в качестве переменной Count. Естественно, этот кирпичик должен быть подсоединен к обоим логическим цепочкам, проверяющим курсорные клавиши.

Куда интереснее обстоит дело с проверкой нажатия на Enter. Нам необходимо проверить условие равенства переменной Count значениям 1 и 2. Для этого создайте еще две логические цепочки: Keyboard > AND > Scene и Keyboard > AND > Game, но так, чтобы сенсор Keyboard являлся общим для обоих цепочек. Измените контроллер AND на Expression [Выражение] и для каждого, соответственно, введите строки count == 1 и count == 2. Для активаторов, в случае с кирпичиком Scene, выберите подфункцию Set Scene [Установить сцену] и в поле SCE: укажите название сцены с самой игрой (у меня SCE: game), а вот для кирпичика Game необходимо выбрать пункт Quit this game [Выйти из игры] (**рис. 4**).



› Рис. 4. Немного запутанно, но работает!

Осталось только добавить проверку нажатия на Escape в главной сцене игры: это должно приводить к выходу обратно в меню. Думаю, здесь вы справитесь и сами, так что цикл уроков по BGE на этом завершается. И хотя созданная нами игра не отличается каким-то изыском и крутизной – рассматривайте ее как плацдарм для более продвинутых проектов. Впрочем, если добавить к ней визуальные эффекты, поработать над оптимизацией и самим игровым концептом, может получится интересная игрушка. Так что дерзайте! LXF

Образовательный пакет GCompris



Дети любят играть – это факт. Вдвойне хорошо, когда игры не простые, а развивающие. **Галина Пожарина** покажет, какую пользу могут извлечь из свободного ПО самые маленькие (и чуть постарше).



Наш эксперт

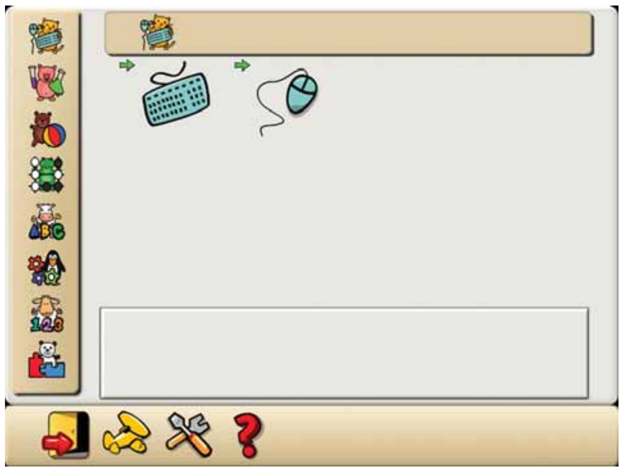
Галина Пожарина
Руководитель Центра информатизации администрации г. Санкт-Петербурга, PhD/DCS

Продолжим разговор об использовании ОС Linux детьми младшего школьного возраста. Графический редактор *TuxPaint*, о котором шла речь в LXF106, может быть установлен как самостоятельная программа (и в данном случае он будет доступен через меню Образовательные приложения), но он также входит в очень интересный пакет *GCompris* (**рис. 1**) – о нем-то мы сегодня и поговорим. Название происходит от французского «J'ai compris» – «Я понял!». Проект существует с 2004 года и локализован на 25 языках, включая русский. *GCompris* – свободное ПО, доступное для Linux и Microsoft Windows, хотя в «целях продвижения GNU/Linux» (так, по крайней мере, утверждается на официальном сайте проекта, gcompris.net) загружаемая Windows-версия ограничена по функционалу и ее «разблокирование» стоит денег. В состав *GCompris* входит более 80 различных игр, сгруппированных по восьми категориям и адресованных прежде всего детям дошкольного и младшего школьного возраста. Игровые персонажи яркие и увлекательные, имеется звуковое сопровождение.



› **Рис. 1. Стартовый экран GCompris.**

Чем же займет ребенка образовательная сюита? В первую очередь, научит пользоваться клавиатурой и мышью (**рис. 2**). «Падающие буквы» и «Падающие слова» необходимо найти на клавиатуре и, естественно, нажать нужные кнопки, игра же «Числа и кубик» при похо-



› **Рис. 2. Пакет научит маленького ребенка пользоваться и клавиатурой, и мышью.**

жем сценарии учит считать и соотносить цифру на клавиатуре и количество точек на гранях кубика. Поскольку количество точек не превышает шести, уровень доступен даже трехлетним пользователям.

Игр, направленных на координацию «глаз–рука» и обучающих соотносить перемещение мыши по коврику и курсора на мониторе, насчитывается пять. Курсор может принимать вид губки – тогда в задачу ребенка входит «стирание» непрозрачных блоков, закрывающих элементы картинки (игры «Двигай и щелкай мышкой» и «Перемести мышь» – по своей сути, это два уровня игры с одними правилами). Игра «Щелкни по мне» ставит условием точное попадание по рыбкам, плавающим в аквариуме.

Click and draw [Щелкни и нарисуй] – хорошо известное упражнение по соединению точек линиями для получения целостного рисунка. Каждая последующая точка-цель выделяется новым цветом. Игра имеет несколько уровней, различающихся по сложности, но работает только с прямыми линиями. Рисовать мышью кривые – смысл еще одной игры, «Полей цветы». Курсор укажет воде путь по шлангу, а его отклонение от заданной линии вернет влагу обратно, и цель не будет достигнута.

Второй блок игр (**рис. 3**) назван авторами «Задания на смекалку». Он включает семь сюжетов, в том числе: прыжки с парашютом,



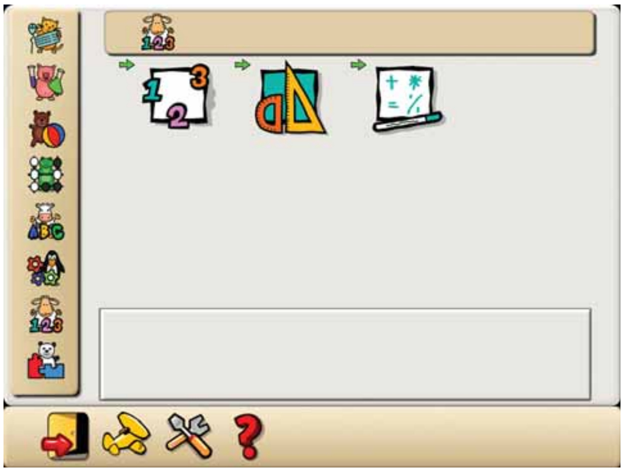
› **Рис. 3. Прыжки с парашютом помогут развить смекалку.**

управление шлюзом, морская регата. Более сложные игры, построение электрической схемы и системы водоснабжения, требуют углубленных знаний об окружающем мире, поэтому рекомендуются пользователям постарше – пяти-семилетним игрокам.

Блок «Развлечения» (**рис. 4**) содержит две графические программы: «Рисуем вместе с Туксом» и «Создай рисунок или мультфильм», текстовый процессор, чат, игру «Футбол» и «Шестиугольник». Две последние могли бы быть отнесены к другим разделам. «Шестиугольник» – логическая игра, напоминающая известную детскую забаву «Тепло-холодно». Ближе к цели поля имеют теплые



› **Рис. 4. Блок «Развлечения».**



› **Рис. 5. Блок обучающих игр.**

цвета («горячо» соответствует красному); чем дальше от цели, тем холоднее они становятся.

В «Футболе» требуется щелкнуть по шару, чтобы придать ему скорость и направление. Чем ближе к центру шара пришелся щелчок, тем медленнее он будет перемещаться.

Блок «Стратегии» включает шахматы и игры «Последний шар», «Четыре в линию», в вариантах на одного и двух игроков, Oware [Вари]. Уровень стратегий доступен для шести-семилетних игроков, цель и правила описаны на русском языке.

По своей направленности близок к стратегиям и следующий блок – «Головоломки». Он включает девять игр: «Пазл», в котором участникам предлагается собрать из частей картины известных художников, «Танграм», цель которой – составить изображение из различных геометрических фигур, «Ханойская башня» двух разных уровней сложности, «Подъемный кран», «Судоку» и прочие.

Еще три блока включают обучающие игры: «Математика», «Упражнения на чтение», «Цвета, звуки и память» (**рис. 5**). Входящие сюда игры могут быть рекомендованы и для работы воспитателям детских садов, логопедам, психологам. Их содержание подходит для детей дошкольного возраста – они развивают память, тренируют различные виды восприятия и т.д. При обучении чтению достаточно интересны такие игры, как «Узнай слово», «Вставь пропущенную букву», «Назови изображение». Богатый арсенал подобных игр активно используется как в детских садах, так и в младшей школе.

Помимо сборника игр, *GCompris* включает панель администрирования. Этот инструмент управления доступен через меню Образовательные приложения вашего рабочего стола, и он дает возможность назначить различным группам пользователей игры различных категорий сложности (**рис. 6**). Таким образом можно настроить пакет для использования детьми разных возрастов, не смешивая уровни и не провоцируя ребят играть в «слишком простые» или «слишком сложные» для них игры. Кроме этого, такие настройки допускают применение *GCompris* не только в домашних условиях, но и в компьютерных классах школ, кабинетах психологов, логопедов. LXF



› **Рис. 6. Административная панель GCompris.**

Любопытство и до GNU/Linux доведет

Город Рубцовск Алтайского края – это своеобразные ворота на торговом пути в государства Средней Азии и Китай. **Тамара Николаевна Воронова** из МОУ СОШ № 1 полюбопытствовала: «А что такое Linux?» Наша история – о том, что из этого получилось.



› **Тамара Николаевна Воронова.**

Евгений М. Балдин (LXF): **Тамара Николаевна, расскажите, пожалуйста, о себе.**

Тамара Николаевна Воронова (ТНВ): Учитель информатики, закончила Новосибирский государственный педагогический университет. Работаю в школе 15 лет.

Как у школьного преподавателя у меня высшая квалификационная категория, и я являюсь победителем ПНПО «Лучшие учителя России» 2006 года.

LFX: **Когда вы впервые услышали о GNU/Linux?**

ТНВ: Слышала я об этой системе и раньше, но где-то в 2003 году начала покупать журналы CHIP и CHIP Special. И там был специальный выпуск, посвященный «Красной шапке». А потом хороший обобщающий материал по операционным системам в «Школьной компьютерре» (была такая замечательная газета, правда, недолго).

Тогда меня это заинтересовало настолько, что на жестких дисках только что полученных компьютеров я «отрезала» по 5 ГБ под будущую установку GNU/Linux. Была мысль показать его группе профильного изучения информатики, с которой мы обсуждали эту тему, и я их заинтересовала существованием других ОС (тогда об этом вообще мало говорили, и слово «Linux» знали немногие).

Идея тогда так и не реализовалась, но дома я ее все-таки поставила! У меня все получилось. Первым было удивление: а в чем принципиальная разница для пользователя? Сейчас я это понимаю: обычный пользователь работать на компьютере не умеет, но Windows дает ему ложное представление о собственном всемогуществе, а Linux обнаруживает правду. А кому хочется услышать про себя «А король-то голый!»

Проблема в том, что школа имеет затягивающий в себя «геймплей» (так, кажется, это называется). Повседневные дела и заботы не дают возможности нормально самообразовываться. Каждый раз себе говоришь: «Вот в субботу я сяду…», «Вот на каникулах я буду…»… Где-то через пару месяцев я обнаружила, что забыла пароль root, а немного погодя «слетела» Windows…

Вернулась к этому вопросу я пару лет назад. Упомянутый журнал CHIP на своих дисках начал размещать различные дистрибутивы, и я снова начала эксперименты на домашней машине. И вот здесь я столкнулась с первой проблемой: как установить дополнительное ПО? Параллельно, на Windows, я устанавливала кросс-платформенное программное обеспечение на своем и на школьных компьютерах, что позволяет мне более уверенно себя чувствовать и в GNU/Linux. Этот пристальный интерес со мной разделил мой коллега Алексей Иванович Мусихин. В этом вопросе он оказался более целеустремленным и достиг больших результатов в деле освоения GNU/Linux. Именно с его помощью среди моих бывших учеников удалось обнаружить несколько линуксоидов.

LFX: **Какое программное обеспечение вы используете в процессе обучения?**

ТНВ: Будете смеяться, но первая Linux-программа, которую я использовала на уроках, был калькулятор (мы с 10-м классом считали объем информации при вероятностном подходе). Я думаю, мой стандартный набор достаточно баналeн: *Mozilla Firefox*, *OpenOffice.org*, *GIMP*, *Gcompris*, *FreePascal* и другие. Конечно, мы не можем так сразу отказаться от программ, работающих в DOS и Windows, поэтому для их запуска используем эмуляторы *DosBox* и *Wine*. О наработках пока говорить рано. Можно говорить о том, что большую часть уже имеющихся материалов можно использовать и дальше без переработки.

LFX: **Другие учителя используют компьютер на своих уроках?**

ТНВ: Предметники используют компьютер, как правило, для подготовки к урокам и для участия в различных конкурсах. Проведение предметных уроков в компьютерном классе сопряжено с двумя основными трудностями: в нашей большой школе всего два компьютерных класса, а информатика изучается с 0 по 11 класс, то есть туда просто не протолкнуться. А кроме того, подготовка такого урока требует весьма значительных усилий и временных затрат со стороны учителя. Если бы была возможность проводить такие уроки в системе, то, возможно, учителя и пошли бы на дополнительные затраты ради повышения эффективности. А эффективность одного урока в четверть/год сомнительна. Решением этой проблемы станет установка комплекта «компьютер+проектор» в предметные кабинеты… Стоимость вопроса: 1,5 млн.рублей.

LFX: **По каким причинам возник вопрос перехода на GNU/Linux?**

ТНВ: Любопытство. Очень хочется разобраться. Мне близка и понятна философия свободного ПО. Очевидно, что одной из основных причин является отсутствие денег. Было бы глупо это отрицать.

Ну и надоело находится в подвешенном состоянии: «придут/не придут», «дадут/не дадут» (не потому что я боюсь наказания, а потому, что

не люблю неопределенности). И вообще, как-то не хочется чувствовать себя нарушителем закона. Тем более что дети начали задавать нелицеприятные вопросы. В этом году тоже спрашивали [*улыбается*]. А еще, хочется дать своим ученикам больше знаний. Что такое Windows и как с ним работать, знают (или думают, что знают) все, а мои ученики теперь имеют возможность «потрогать руками» Linux и сравнить. У них есть выбор. А это всегда хорошо.

Дополнительным бонусом от перехода было полное решение проблемы безопасности. Слова «вирус» и «антивирусная программа» мы не забыли, но это уже к нам не относится. Меня больше не пугает перспектива установки компьютеров в библиотеке и/или учительской.

LFX: Был ли план миграции?

THB: Плана как такового не было. Был вынужденный порядок, так как переход начался в сентябре (как вы понимаете, не самый удачный месяц для начала глобальных изменений). Поэтому в первую очередь перевели ученические машины в компьютерном классе. Затем наступила очередь администрации. На двух машинах GNU/Linux поставили вторую ОС, так как на них стоит лицензионная Windows, и мы решили, что будет полезным, если на паре машин она останется.

Очень хотелось поставить Linux на наши старенькие машины (P-I 166MMX, 16 Мб ОЗУ), но у нас ничего не получилось. Мы оставили эту затею, чтобы позже попробовать сделать из них терминалы.

LFX: Страшно не было?

THB: Страшно было очень, все было страшным. Как это будет? Не сорвем ли учебный процесс? Сумеем ли пройти программу? Как отреагируют ученики и учителя?..

На самом деле, не знаю, как я решилась на такую авантюру (я не одна принимала и реализовывала это решение, но говорю сейчас именно про себя). В итоге: учебный процесс не сорвали, программу проходим нормально, ученики и учителя проявляют сдержанный интерес. Нам удалось избежать главной проблемы – отсутствия драйверов (три принтера и два сканера заработали). Единственно, мы не делали попытку «завести» под Linux интерактивную доску, так как оборудование специфичное и поставлялось с лицензионным ПО. И при покупке новых устройств приходится оглядываться на эту традиционную для GNU/Linux проблему.

LFX: Какой был выбран дистрибутив GNU/Linux в качестве базового и почему?

THB: Мы выбрали Ubuntu 7.04, потому что он очень дружелюбен. Именно в Ubuntu я поняла, как ставить дополнительные программы [*улыбается*]. А еще потому, что с дистрибутивами проблем нет, есть специальный образовательный дистрибутив, сравнительные анализы того периода (**LXF** как-то публиковал, а к тому времени я уже покупала ваш журнал) говорили в пользу Ubuntu.

LFX: На все это хватает собственных сил или кто-то помогает?

THB: Собственных сил пока не хватает. Сейчас нам помогает выпускник нашей школы Владимир Владимирович Черников. Говорить о LUG в г. Рубцовске не приходится, но есть выпускники, к которым можно обратиться за помощью. В большей степени были бы полезны качественные курсы, как в Томске, например.

Хотелось отметить, что когда говорят об Ubuntu, то в первую очередь хвалят его сообщество. Его существование значительно облегчает жизнь.

LFX: Как на GNU/Linux отреагировали учащиеся?

THB: Было смешно. Все спрашивали: «а это Vista?» По большому счету ученики никак не отреагировали. Хотя о том, что «это не Vista» знают, наверное, все. Кто-то из старшеклассников расспрашивает подробнее, кое-кто просит записать дистрибутив на диск, но особого ажиотажа нет. Мне кажется, рано говорить об использовании GNU/Linux учениками дома. Мы будем над этим работать. Например, через пропаганду кроссплатформенного ПО.



› **Владимир Владимирович Черников: он помогает поддерживать класс.**

LFX: Были ли проблемы при переходе у учителей и администрации?

THB: Не было у нас проблем. После того как пошла волна по лицензированию ПО в школе, наш директор (Любовь Николаевна Гузеева), выяснив возможные пути решения проблемы, поддержала идею миграции на Linux. И учителя отреагировали нормально, только периодически спрашивают, когда будут курсы по новой «оси».

Я в одной статье прочитала мысль о том, что при полностью настроенном компьютере пользователю безразлично, какая ОС у него стоит – полностью согласна. Наш опыт это подтверждает.

LFX: Что изменилось после получения «Первой помощи 1.0»?

THB: «Первую помощь» получили только в начале апреля. Появилась опасность, что заставят это все ставить на компьютеры. Надеемся, что нет. У самих такое желание не появилось. Нам просто передали коробку и сказали «Разбирайтесь!» На следующий день спросили «Ну, как?» – и мы ответили, что большую часть можно просто выкинуть. На этом пока все и закончилось.

Польза от «Первой помощи» одна: на ноутбуке стояла лицензионная Vista – поставили Windows XP. Он постоянно в работе, и нельзя его на одну-две недели вывести из строя, подбирая подходящий дистрибутив с поддержкой всего оборудования. Да! Еще *FineReader*. Без него было тяжко.



› **Факультатив по *GIMP*.**



› А.И. Мишин и *Gcompris*.

У Коэльо, кажется, читала мысль: «Когда ты открыто объявляешь о своих принципах, тебе приходится жить в соответствии с ними». Мы открыто объявили о переходе на Linux, а теперь в кусты? Не пойдет. Это раз. Кроме того, все лицензии до декабря 2010 года. Практически всего на два с половиной года, а потом у нас будет «дежа-вю». Потому что нужно будет ставить Linux и начинать все сначала. Неразумно. Это два. То есть если администрация будет настаивать, то на их машины установим две системы, но мне кажется, не стоит делать и этого. В компьютерных классах я постараюсь этого не допустить.

LFX: Поменялось ли что-нибудь в образовательном процессе из-за перехода на GNU/Linux?

THB: Образовательный процесс после перехода на СПО, мне кажется, никак не изменился. И это плюс. Это касается всех участников процесса.

Правда, замечу, что у меня за весь год ни на одном компьютере не поменяли картинку на рабочем столе [*улыбается*]. В Windows это жутко раздражало. Отказы машин происходят только по причине аппаратных сбоев. Хотя Linux и более чувствителен к различным нарушениям (например, «флэшка» или диск с нарушениями файловой системы не монтируется вообще).

LFX: Все вернуть назад не хочется?

THB: Желания откатиться не возникло. Я не хочу зависеть от действий и решений министерских чиновников. Моя жизнь сложилась так, что я не привыкла ждать у моря погоды. Если могу решить проблему своими силами, то делаю это. А если нет, то считаю проблему нерешаемой. В случае с ПО тоже так было. Из всех возможных вариантов выбрала установку GNU/Linux и закрыла этот вопрос навсегда. На чем основаны надежды остальных? Не знаю. На русском «авось», вероятно. Как говорил первый директор: «А в первой школе будет так».

LFX: Какие методические материалы сейчас используются?

THB: Используем в основном собственные наработки. Хотя это мало связано с выбором операционной системы и конкретных программных продуктов. Слежу за публикациями в **Школе LFX**.

LFX: Что-нибудь на будущее запланировано?

THB: К лету мы планируем закончить модернизацию одного из компьютерных классов, чтобы установить там ту же ОС. Закончим создание школьной сети, в том числе, с использованием беспроводных технологий. В условиях нехватки компьютеров хочется сделать терминалы из старых машин (сервер у нас уже есть, работает под управлением Edubuntu). Потребность такая есть, и очень серьезная, так как в нашей школе внедряется NetSchool, а новых компьютеров не предвидится. Еще хочется завести всем пользователям свои учетные записи и разработать грамотную разрешительно-запретительную политику.

LFX: Чем, на ваш взгляд, Linux удобнее Windows?

THB:

» Идеология открытого ПО. Для меня это важно, теперь все разговоры о покупке лицензии государством «по барабану».

» Технология дистрибутива (когда вместе с операционной системой сразу получаешь все необходимое). Особенно, если учесть, что можно освоить технологию создания собственного дистрибутива, дополнив основной парой-тройкой нужных тебе программ. Хотя можно поступить проще – доставить из репозитория (тоже хорошее изобретение) нужные программы и слить их на отдельный диск.

» Безопасность работы. Ви-ру-сы! Ау… Плюс автоматические обновления, исправляющие уязвимости.

» Дети ничего не могут сделать (администрирование закрыто) и ничего не могут принести. Напоминаю, что в Windows очень много программ, которые не работают под ограниченной записью, только под администратором, а давать детям права администратора… чревато, я бы сказала.

» Сетевая политика, ограничивающая пользователя не только в настройке, но и запуске «ненужных» программ по его имени и/или группе. Это еще не сделано, но я знаю о такой возможности и мы будем ее реализовывать.

LFX: Чего вам сейчас не хватает для полного счастья?

THB: Для полного счастья мне не хватает времени. В том числе и для того, чтобы полнее разобраться с ОС. То есть мне еще и знаний не хватает. Кроме этого, до сих пор не решены две проблемы: распознавание текста и редактирование видео. Видимо, придется купить эти две программы. Вот, вроде бы, все.

LFX: Чего бы вы пожелали сообществу, чтобы процесс внедрения GNU/Linux в образование пошел активнее и как минимум не откатывался назад?

THB: Терпения. Прокуратура и Microsoft сделают все остальное.

Где-то вычитала фразу: «Очень жаль, что главным рекламщиком Linux является Windows». Еще пара громких дел по типу Поносова – и все будет в порядке. Во всем мире GNU/Linux используется в школах, в том числе из-за недостатка финансирования. У нас с этим все на мировом уровне (школы финансируются из рук вон плохо). Пройдет лет 5, и в большинстве школ будет СПО. Тратить деньги на проприетарное ПО директора не будут, лучше купить новые парты. А нелицензионное использовать уже небезопасно.

Необходима более качественная локализация deb-пакетов. Чтобы «из коробки» ставилось больше программ (так как бесплатный Интернет в школах скоро закончится). Старая проблема: кодеки, она по-прежнему отпугивает новичков. В дистрибутивах для России их нужно ставить по умолчанию. Если я правильно понимаю, у нас это не является нарушением законодательства.

Есть пожелание и «из области фантастики»: пусть каждый линуксоид придет в свою школу и проведет семинар-практикум на тему «Linux в школе: плюсы и минусы» и поможет поставить/настроить его в качестве второй ОС на школьных компьютерах. Да хотя бы просто придет в школу по соседству с домом или работой и скажет: «Я имею опыт установки и настройки GNU/Linux. Вот мои координаты. Если нужна будет помощь в этом вопросе, то обращайтесь».

LFX: Вы упомянули, что следите за школьной тематикой LXF. Что вы можете сказать о самом журнале? Есть ли пожелания?

THB: Мне сложно сейчас говорить о качестве журнала: я просто еще не все понимаю. Читаю выборочно: интервью, обзоры и характеристики ПО… Например, очень понравился материал о различных языках программирования. А журналу можно пожелать расширить раздел для начинающих и/или учителей. LXF


КОРПОРАТИВНЫЕ СЕРВЕРЫ
И СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ
(812) 327-5960
(495) 232-9230
info@trinitygroup.ru
TRINITY
CORPORATE IT PROJECTS
Анализ существующей IT инфраструктуры
Разработка технического задания
Проектирование, монтаж, внедрение
Комплексное управление IT инфраструктурой
Катастрофоустойчивые решения
Серверы
под Linux
FreeBSD
Solaris x86
для баз данных, интернет шлюзов, WEB-приложений, кластеры для научных расчетов
Мы делаем бизнес успешным
www.trinitygroup.ru
Информационные технологии
от экспертов


СИСТЕМНЫЙ
администратор
Клонируем Windows
с помощью Symantec Ghost
Насколько неуязвима ваша
беспроводная сеть?
Active Directory вместо
рабочей группы
Настраиваем DSPAM –
ваш личный спам-фильтр
Как спасти данные,
если отказал жесткий диск
Модифицируем BIOS
Все ли возможности ClamAV
вы используете?
Что важно знать
об IP-телефонии
Админские сказки
www.SAMAG.ru
В «Системном администраторе»
вы не прочтете о:
– котировках валют
– сплетнях
– погоде
– политике
– развлечениях
В вашем распоряжении:
– опыт лучших IT-специалистов
– новые идеи и полезные советы
– самые эффективные решения
в области системного и сетевого
администрирования
Подпишитесь сейчас!
Время подписки ограничено!
Роспечать – 20780, 81655
Пресса России – 87836
Online-подписка – www.linuxcenter.ru



» Записал Евгений Балдин



История успеха

» **Кто**
Александр Геннадьевич Фенстер, математик-программист, преподаватель информатики.

» **Откуда**
Механико-математический факультет Новосибирского государственного университета (НГУ).

» **Проект**
Адаптация курса информатики для выполнения практических занятий в среде GNU/Linux.

» **Результат**
Технология опробована, студенты освоились,

GNU/Linux проникает в НГУ

Чтобы удачно внедрить GNU/Linux повсеместно, одобрение руководства, как правило, необходимо. А как быть, если руководству в лучшем случае все равно? Можно ли что-то сделать своими силами? **Александр Геннадьевич Фенстер** показывает одну из возможных дорожек в нужном направлении.

Преподаванием программирования я заинтересовался, еще будучи студентом старших курсов. Начал вести занятия в Высшем колледже информатики при НГУ и в Летней школе юных программистов. Сейчас в свободное от основной работы время преподаю в самом университете на механико-математическом факультете.

До университета вообще ни разу не работал в системах, отличных от MS-DOS и Windows. На первом курсе ММФ НГУ пошел на факультатив по языку программирования LISP. Преподавателем был сотрудник Института систем информатики СО РАН, который создал всем учетные записи на своем рабочем компьютере (там был установлен Debian Potato). Как сейчас помню листочек блокнота с выписанными командами *ls*, *ps*, *kill* и чем-то еще таким. Впечатление было, конечно, то еще: как будто пытаешься объясниться на иностранном языке, которым не владеешь. Впрочем, это ощущение быстро прошло. Затем как-то так получилось, что я перенес свою почту на этот сервер и начал читать ее программой *pine*, потом перевез ICQ (сначала использовал *licq*, затем *zicq*, *centericq* и сейчас *centerim*). Приглянулось, что почта и «аська» доступны независимо от того, где я сам физически нахожусь: я просто под-

ключался по SSH к серверу и работал. Где-то в это время уже было несколько машин, на которых у меня были права администратора. Затем я попробовал использовать GNU/Linux на своем рабочем столе – и понравилось. Сейчас с одинаковой частотой использую две системы: Debian Lenny дома и лицензионный Windows XP на рабочем ноутбуке (от корпоративных стандартов никуда не денешься).

Курс программирования состоит из лекций, семинаров и практики. На лекциях студентам читаются общие вещи, касающиеся языков программирования и алгоритмов, без привязки к конкретным компиляторам и операционным системам. На семинарах (в обычных аудиториях, без компьютеров) изучается язык C и решаются задачи по программированию и разработке алгоритмов. На практике в компьютерном классе студенты решают и сдают 15 семестровых заданий. В конце семестра студенты сдают зачет и экзамен.

От семинаров и практики требуется, чтобы студенты по окончании курса знали синтаксис C и могли самостоятельно писать код: например, реализовывать вычислительные методы в курсе вычислительной практики. У меня есть свой набор задач, который сформировался за несколько лет преподавания. Ознакомиться с ним, а также с информацией для студентов, можно на сайте **7113.fenster.name** (7113 – это номер студенческой группы).

Раньше на первых практических занятиях я показывал основы работы в Microsoft Visual Studio, и дальше студенты возились с этим огромным монстром, используя от силы 5 % его возможностей. С моей точки зрения, учить писать простые (100–200 строк максимум) программы с использованием этой технологии – это как стрелять из пушки по воробьям. Куча времени уходит только на обучение самой среде: как создать проект и как запустить программу. Для начинающих все это совершенно не нужно и только отвлекает от главного – а именно от написания кода. В этом году мне захотелось перейти на более естественный и легкий инструментарий, и среда GNU/Linux выглядела для этого как нельзя более подходящей.

К сожалению, условий для такого рода изменений не было никаких, так как в терминальных классах нашего факультета везде установлена ОС Windows XP, без каких-либо исключений. Однако все оказалось не так страшно, как представлялось. Для организации доступа студентов к среде GNU/Linux нужно было найти сервер и выбрать SSH-клиент. Также перед началом обучения необходимо было определиться с набором базовых программ.

1 Выбор сервера. Сначала попробовал пойти официальным путем – зайти к администраторам и попросить создать для студентов учетные записи (или хотя бы одну на всю группу). Т.к. просьба оказалась беспрецедентной для мехмата, то мне сказали, что будут говорить с главным админом и решать. В итоге админ «нарезал» для меня виртуальную машину на базе OpenVZ в сети НГУ (доступна из НГУ, общежитий и местных локальных сетей), где я и «завел» всех студентов. Подозреваю, что если бы не личное знакомство с администратором, то ничего сделать не получилось бы.

2 SSH-клиент. В качестве клиента SSH я выбрал PuTTY, доступный по адресу **www.chiark.greenend.org.uk/sgtatham/putty/** [и, кстати, являющийся свободным ПО, – *прим. ред.*]. На мехмате НГУ Windows переставляют из образов чуть ли не каждую неделю, а официально внести PuTTY в список нужного для установки ПО я на начало семестра не успел. В итоге придумал простой вариант: на своем сайте (**7113.fenster.name**) разместил ссылку на **.exe**-файл PuTTY, лежащий в НГУ'шной сети. Очень удобно, что инсталляции он не требует. Процесс работы такой: студенты идут на сайт (они в любом случае идут туда читать задачи), щелкают по ссылке, скачивают программу в 400 КБ, запускают, затем вводят адрес сервера, имя пользователя и пароль – вуаля! Быстрее, чем *Studio* запустить.

3 Выбор программ. Подумав, решил остановиться на варианте *bash* + *vim* + *make*. Почему, скажем, не *emacs*? В основном потому, что сам использую *vim* и хочу учить тому, что сам знаю достаточно хорошо. Решил не навешивать никуда никаких псевдонимов, а заставить писать команды полностью: думаю, так полезнее. Программы *mc* и *mcedit* тоже показал, но намного больше внимания уделил *vim*.

Код компилируется при помощи GCC. Некоторые студенты из особо продвинутых разобрались с отладчиком *gdb*.

4 Документация. Написал и выложил краткую инструкцию по командам на русском языке: **7113.fenster.name/commands.pdf**.

Адаптация студентов к незнакомой для них среде прошла проще, чем я ожидал, хотя до начала семестра с GNU/Linux никто из них не сталкивался (я специально это уточнил).

На первом занятии я потратил 40 минут на объяснение нескольких базовых вещей: перемещение по каталогам, использование текстового редактора и команды *make* для компиляции программ. Добился, чтобы каждый студент написал, откомпилировал и запустил «Hello, world!». К моему удивлению, этого оказалось вполне достаточно, чтобы студенты начали работать самостоятельно.

Судя по выводу команды *last*, студенты активно работали и из дома (из общежитий). Был очень приятно удивлен, когда некоторые студенты начали сдавать задачи на своих личных ноутбуках, на которых был установлен GNU/Linux. Здесь, конечно, сыграло свою роль наличие расположенного рядом, в пределах кампуса, зеркала Debian: **debian.nsu.ru**.

Сам курс практически не изменился. Я считаю, что начинать изучение программирования всегда следует с простых программ (консольных приложений), и нет разницы, в какой операционной системе это делать. Так что везде, где есть возможность дать студентам входы на Linux-машину, можно без особых сложностей отказаться от использования громоздких IDE, установленных локально, и перейти на удаленный вариант работы.

По опыту этого семестра можно отметить, что исчезли забытые дома «флэшки» с программами (ведь сейчас все делается на сервере), а также проблемы с созданием, сохранением и открытием проектов в *Visual Studio*. Также появилась возможность заменить некоторые задачи более приближенными к реальности: например, вместо какого-нибудь стандартного задания на обращение с файлами и строками можно попросить разобрать формат **/etc/passwd**. В то же время, работы с моей стороны немного прибавилось: необходимо было написать инструкции, настроить виртуальный сервер и завести учетные записи.

Другие преподаватели к моим начинаниям относятся с интересом, и я надеюсь к следующему году расширить эксперимент на несколько групп. У меня есть лето, чтобы придумать, как перевести следующий курс, где изучается создание графических интерфейсов с использованием *Borland Developer Studio*, на свободные продукты. Если получится придумать адекватную по сложности замену, то это будет просто здорово.

По окончании семестра можно с уверенностью сказать, что эксперимент прошел удачно. Собственно говоря, в этом нет ничего удивительного, так как GNU/Linux давно пригоден для работы.

Основная же, с моей точки зрения, проблема для такого рода переходов, как и десять лет назад, такая: чтобы нормально разобраться с системой, нужно иметь под рукой «гуру», или же достаточное количество терпения для чтения документации, причем чаще всего на английском языке. Если вы готовы тратить часы на чтение «манов» – вы сделаете все, что вам нужно. Но пользователь хочет иметь систему, работающую «из коробки», и хотя каждый новый выпуск того же Debian'а в этом смысле становится все лучше и лучше, до идеала, мне кажется, еще далеко.

В принципе, роль того самого «гуру» может эффективно играть сетевое сообщество. Например, перед самым началом эксперимента я задал вопрос на местном Linux-форуме (**www.nsu.ru/phpBB/viewforum.php?f=9**) и получил несколько достаточно дельных советов.

Не надо бояться экспериментировать. Многие вещи, которые в вузах традиционно выполняются в других ОС, можно делать при помощи GNU/Linux, а если не видно разницы – … LXF



Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru

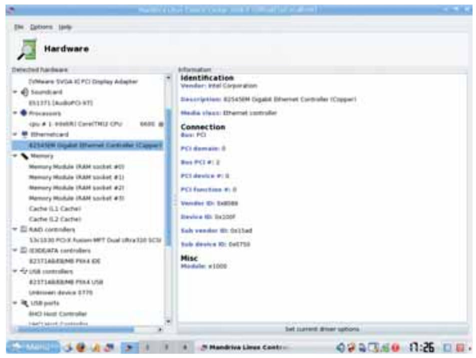
1 Внутренняя информация

Я в основном работаю в окружении Windows, но постоянно стараюсь перейти на Linux; одна отсутствующая в нем вещь разочаровывает меня – это хорошая видимость используемых устройств и их статус с привязкой к драйверам. Стесняюсь и сказать, но менеджер устройств Microsoft – хорошая работа. Если ли в Linux что-нибудь, предоставляющее похожую информацию?

Дэвид Хилл [David Hill]

Данная ситуация в Linux в корне отличается: большинство драйверов включены в ядро, и нет большой нужды сравнивать установленные и запущенные драйвера с доступными на сайтах. Если вы будете регулярно обновлять ваш менеджер пакетов, он будет информировать вас о новых версиях. Есть также множество программ, сообщающих о статусе оборудования – частью стандартных, а частью специфичных для дистрибутива. Одна из моих фаворитов – *lshw* (http://ezix.org/project/wiki/HardwareLiSter); она в основном используется в консоли и выдает детальный список всех частей компьютера, от материнской платы до процессора и USB-устройств. По умолчанию вывод делается в виде простого текста, но можно также сгенерировать HTML для просмотра в браузере или открыть окно, кликнуть на элемент и получить больше информации. Программа имеет ряд опций для ограничения выводимых данных, например, по типу устройств, или удаления из вывода конфиденциальной информации вроде серийных номеров.

Похожая программа – *HardInfo* (http://hardinfo.berlios.de) – отображает в графическом режиме множество сведений об оборудовании и программах. Она выводит информацию в виде дерева, и вы можете перейти к интересующим вас подробностям. В программе есть секция, показывающая загруженные



› **Control Centre в Mandriva – одна из многих программ, доставляющих информацию о вашем оборудовании.**

модули ядра, так что можно увидеть, какие драйверы использует ваше оборудование. Эти программы лучше запускать от имени суперпользователя (root) или через *sudo*, чтобы обеспечить считывание всей возможной информации с вашей системы.

Основные окружения рабочего стола имеют свои программы: *Gnome Device Manager* и *KInfoCentre* в KDE, предоставляющие похожие сведения. Различные дистрибутивы также имеют собственные варианты подобных приложений: *Device Manager* в Ubuntu (он ближе всех к программе в Windows, по крайне мере, мне так показалось при его использовании), *Yast* в SUSE и Control Centre в Mandriva – все предоставляют информацию об оборудовании. Программы от SUSE и Mandriva интегрированы во всеобъемлющие административные оболочки, поэтому они также имеют опции для настройки оборудования там, где это возможно. **НБ**

Наши эксперты

» Мы найдем эксперта по любому вопросу – от проблем с установкой системы или модемом до сетевого администрирования; главное – спросить!



Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс
Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Пол Хадсон
Пол – местный суперпрограммист, и он может и хочет управиться со всеми вашими проблемами по части web и баз данных.



Грэм Моррисон
Когда он не обозревает кучи програмного обеспечения и не халтурит с MythTV, Грэм готов дать ответ касательно любого оборудования и проблем виртуализации.



Валентин Синицын
В редкие свободные минуты главный редактор нашего журнала обычно запускает mcedit, Чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.



Юлия Дронова
Когда ее компьютер не занят выполнением команды emerge, Юлия использует его, чтобы модерировать ЛинуксФорум.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:
Пишите нам по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

2 Мучения мигранта

Я установил Ubuntu с DVD из LXF104, и все шло хорошо. Я был восхищен интерфейсом пользователя и сумел наладить работу с сетью. Настройка Nvidia и принтера также прошли легко и непринужденно. Увы, этим все и закончилось.

Я решил установить кое-какое ПО и начал с *FlightGear*: распаковал файлы и попытался найти инструкцию по установке или, как я делал в Windows, установочный EXE-файл. В конце концов я нашел инструкции – явно рассчитанные на программистов – говорящие, как скомпилировать программу. Последовал этим инструкциям буквально в командной строке, но в итоге получил только кучу ошибок.

Я не сдался и попробовал выполнить инструкции по установке *Toribash*. Первая часть работала, но когда я напечатал **toribash_ubuntu7** в командной строке, то получил ошибку *“bash: toribash_ubuntu7; command not found”*.

Я бы с радостью выбросил или по крайней мере »

» отложил бы в сторонку Windows, но если в Linux так сложно устанавливать программы, то, боюсь, еще пару лет Windows у меня поживет. Описания всяческих программ в вашем журнале фантастические, но какая в них польза, если нормальному человеку их не поставить? Почему ни один из создателей программ, вложив в них огромную энергию, ум и самоотверженность, не мог написать простой **install.exe** под Linux для тех, кто пытается влиться в Linux, или придумать другой механизм?

Алистер Гилмор [Alastair Gilmore]

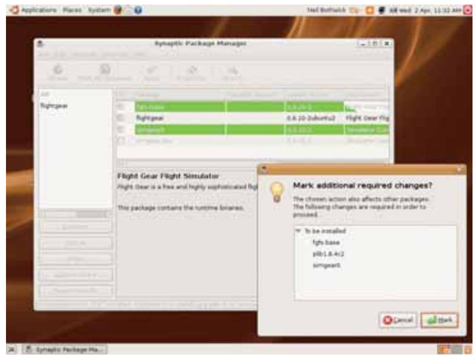
О Одна из самых больших проблем при исследовании новой ОС – это необходимость «отучить» себя идти привычным путем. Linux – не Windows, и многие вещи в нем сделаны иначе; инсталляция ПО – один из самых ярких примеров. Существует три основных пути установки программ в Linux: компиляция из исходных кодов, загрузка пакетов с сайта программы (или DVD) и установка через менеджер пакетов вашего дистрибутива. Второй способ близок к подходу Windows, хотя и не применяет исполняемых файлов – пакет содержит все необходимое и загружается с помощью специального менеджера. Если доступен Deb-пакет, установите его с помощью команды

```
sudo dpkg --install someprogram.deb
```

Этот способ работает, но страдает от некоторых ограничений, как и метод Windows. Вам надо заново посетить сайт и узнать о наличии обновлений, возможен конфликт с другим установленным ПО, и у вас нет информации о целостности загруженного пакета. Всего этого можно избежать, используя менеджер пакетов дистрибутива и репозитории. Репозиторий – это коллекция пакетов, собранных и протестированных для вашего дистрибутива и проверенных на отсутствие проблем с безопасностью. Пакеты подписаны цифровой подписью и проверяются менеджером пакетов, что обеспечивает «чистоту» загружаемого ПО. Это не только лучший способ установки программ, но также и самый удобный, и включает все, что вы хотите. Просто запустите *Synaptic* (в случае Ubuntu), нажмите кнопку Search [Поиск], выберите из найденного то, что вы хотите установить, и нажмите Apply [Применить]. Менеджер заботливо отыщет необходимые пакеты, включая любые зависимости (пакеты, необходимые вашему для работы), скачает и установит их, а также проинформирует вас о любых доступных обновлениях.

Некоторые графические менеджеры пакетов, например, *Yast* от SUSE, могут выполнять установку из скачанных вами пакетов или найденных на DVD, но *Synaptic* пока этого не делает, так что без *dpkg* не обойтись.

Если вы хотите скомпилировать программу из исходных текстов, вам понадобится пакет **build-essentials** – установите его с помощью *Synaptic*. Ошибка *Toribash* произошла потому, что Linux может искать команды только в списке специальных директорий, который из соображений безопасности исключает текущий каталог. Для запуска команды, находящейся в данной директории, применяется префикс **./**, т.е. **./toribash_ubuntu7**. **MC**



› **Установку ПО в Linux лучше делать с помощью менеджера пакетов – обычно он располагает десятками тысяч доступных приложений.**

3 Конфуз UUID

В У меня Ubuntu 7.10, и я задумал переразбить мой жесткий диск и перенести **/home** на новый отдельный раздел, как было описано в статье Джека Найта в LXF100/101, но без шифрования. Но тут возник ряд вопросов, и прежде чем совершить этот поступок, я хотел бы посоветоваться.

Из статьи, а также из тех книг, которые я прочел, непонятно, как дать знать операционной системе, где находится новый раздел **/home**. Инструкции в статье применимы к ситуации, когда процесс шифрования завершен.

По-моему, ответ должен лежать в **fstab**, поэтому я заглянул в него (прилагаю копию) и обнаружил, что существующий раздел Linux имеет запись UUID, которая, откровенно говоря, непостижима. Нормальные записи **fstab** для двух разделов Linux, похоже, закомментированы:

```
# /dev/sda3
UUID=ff773431-fb57-48b4-bb55-01da6902c372 /
ext3 defaults,errors=remount-ro 0 1
```

Если я запускаю *GParted* из меню System/ Administration, то не могу изменить размер разделов Linux – я думаю, потому, что они смонтированы, и это очень опасное занятие – редактировать смонтированные разделы. А вот Live CD-версия *GParted* позволяет редактировать любой раздел на моем компьютере.

Джон Пейтон [John Paton]

О Вы на правильном пути в попытках добавления нового **/home** в **/etc/fstab**. Запись обычно выглядит похожей на

```
/dev/sda5 / ext3 defaults,errors=remount-ro 0 1
```

Но, как Вы обнаружили, Ubuntu использует UUID вместо номеров разделов. Закомментированная строка просто показывает, к чему относился UUID в момент установки. UUID – это уникальный идентификатор, применяемый к файловой системе при ее создании, и он не меняется в течение всей ее жизни. Если вы хотите ужать **/dev/sda2** и добавить другой раздел между ним и текущим **sda3**, то последний будет изменен на **sda4**, и обычный **fstab** уже не будет работать, а **fstab** в стиле Ubuntu останется действительным благодаря сохранению все того же UUID.

Для добавлении новой домашней файловой системы есть несколько вариантов. Можно сделать это известным вам способом, используя стандартный метод **/dev/xxx** и зная, что при перемещении разделов понадобится редактировать **fstab**. Или можно пойти »

! Вопрос-победитель (русская версия)

Юрий получает подарочный сертификат на 1000 рублей от интернет-магазина ГНУ/Линуксцентр! Просим победителя выйти на связь с редакцией: info@linuxformat.ru

★ И тишина была ему ответом

В После изнурительного выбора дистрибутива, способного удовлетворить мои потребности, я остановился на двух: ASPLinux 11.2 (клон Fedora Core 4) и Fedora 8. Именно эти две системы оказались способны способны работать с моей видеокартой (ATI Sapphire Radeon X1950 Pro с чипом AGP) в режиме VESA. Однако я хотел бы установить в них проприетарный драйвер, чтобы можно было играть в игры.

Я долгое время общался с участниками разных форумов, но попытки помочь мне не увенчались успехом. В ASPLinux установить драйвер не получается вообще. В Fedora, наконец, он стал устанавливаться без ошибок, но после изменения конфигурационного файла и перезагрузки я наблюдаю черный экран. Не работает клавиатура. Нет курсора мыши. Если в конфигурационном файле заменить **fglrx** на **vesa**, то можно вернуть графический режим, но для меня это не выход.

Юрий Клещетников

О Похоже, Вам повезло увидеть печально известный «Черный экран смерти» от ATI. В данном случае, причина его – ошибка, связанная с несовместимостью модулей *edac* и *agpgart*, в результате чего AGP-версии карты серии ATI Radeon X1k/X2k (как ваша X1950) с чипсетами PCI-Express и мостом AGP неверно определяются драйвером как карты PCI-E.

Чтобы попробовать изменить ситуацию к лучшему, нужно занести в черный список модули *i82875p_edac* и *edac_mc*. В Fedora для этого достаточно отредактировать (от имени root) файл **/etc/modprobe.d/blacklist** и внести в него следующие строки:

```
blacklist i82875p_edac
blacklist edac_mc
```

Далее, попробуйте перезагрузиться. Это решение работает во многих дистрибутивах, возможно, оно поможет и вам. Если загрузка пройдет успешно, проверьте как работает драйвер командами *fglrxinfo* и *glxinfo | grep direct*.

Как вариант, можно попробовать поставить ATI-драйвер версии 8.2. Известно, что с картами ATI Sapphire Radeon X1950 Pro он работает хорошо. Для загрузки драйвера этой версии пройдите по ссылке: http://ati.amd.com/support/drivers/linux/previous/linux-rf-cat82.html Вы также можете найти его на LXFDVD в разделе **Главное**.

ЮД

» путем Ubuntu, используя команду *vol_id* для получения UUID нового раздела.

```
$ sudo vol_id /dev/sda5
ID_FS_USAGE=filesystem
ID_FS_TYPE=reiserfs
ID_FS_VERSION=3.6
ID_FS_UUID=e242a0ee-f07e-45f2-a104-c8603ccfbe04
ID_FS_UUID_ENC=e242a0ee-f07e-45f2-a104-c8603ccfbe04
ID_FS_LABEL=
ID_FS_LABEL_ENC=
ID_FS_LABEL_SAFE=
```

Здесь вы можете увидеть UUID для файловой системы и скопировать его в **fstab**. Есть и третий вариант, и вывод *vol_id* дает вам подсказку – метка файловой системы; это метод, предпочитаемый Red Hat/Fedora. Как и UUID, метка не меняется при добавлении разделов, но она легче читается. Все, что вам нужно – дать метку вашему разделу командой

```
e2label /dev/sda5 HOME
```

затем отредактировать **/etc/fstab** так:

```
LABEL=HOME / ext3 defaults,errors=remount-ro 01
```

Сменить метку существующей файловой системы ext3 можно с помощью *e2label* без нарушения содержимого, так что вы можете дать имя своему корневому разделу и исправить **fstab**. Файловые системы, отличные от ext3, все имеют свои собственные инструменты для задания метки тома, вы даже можете пометить свой раздел swap с помощью *mkswap -L* ....

Вы правы, что *GParted* не работает со смонтированными разделами, но вам не нужен отдельный LiveCD; можно загрузиться с установочного диска Ubuntu и запустить его оттуда. **НБ**

4 Аквариум на рабочем столе

В У меня запущен Gnome в Mint 4.0, дистрибутиве на базе Ubuntu 7.10. Можно ли установить хранитель экрана *Sherman's Aquarium*? Я установил его, и он запускается как апплет панели Gnome, и я могу вручную стартовать большую версию из командной строки, но он не появляется как хранитель экрана в соответствующем списке. Похоже, что последний берется из конфигурационных XML-файлов, но я не хочу с ними возиться, не представляя, как это делать.

liegerm, с форумов

О *Sherman's Aquarium* работает с *XscreenSaver*, но не с *Gnome-screensaver*. *XScreenSaver* по умолчанию не установлен в Linux Mint, поэтому первым шагом будет его добавление. Установка программы *Sherman's Aquarium* обычным путем (через менеджер пакетов *Synaptic*) не включает ее в список хранителей экрана *XScreenSaver* – возможно, потому, что этот способ считается «левым».

Для добавления в список нужно отредактировать файл **.xscreensaver** в вашей домашней директории. Если он не существует, запустите *xscreensaver-demo* для его создания с настройками по умолчанию. Теперь откройте файл и найдите в нем строку **programs:**. Добавьте после нее:

```
"Sherman's aquarium" shermans -root \n\
```

Затем запустите *xscreensaver-demo* для настройки хранителя экрана. Наконец, Вам надо убедиться, что *XScreenSaver* запускается при старте сессии, и отключить *Gnome-screensaver*. Зайдите в System > Preferences > Sessions, нажмите Add, напечатайте соответствующее имя и описание и установите команду в

```
xscreensaver -nosplash
```

Отключите *Gnome-screensaver*, выйдите и зайдите снова, и Ваш рабочий стол будет кишеть рыбками. **ГМ**



› ***Sherman's Aquarium*** **– отличный хранитель экрана, но его можно использовать только с *XScreenSaver*: он не работает с установками Gnome по умолчанию.**

5 Широкополосный беспроводной модем

В Я купил *Linux Format* Февраль 2008 с Mandriva 2008 [LXF102]. Я использовал продукты Microsoft 25 лет и должен признать, что Linux в корне отличается и по-настоящему требует иных решений. Но все равно числю себя под знаменами «L». На моем настольном компьютере я пользуюсь беспроводным широкополосным модемом Maxon модели BP3-EXT для доступа в Интернет из Windows XP. Мой провайдер Bigpond в Австралии проинформировал меня, что не поддерживает операционную систему Linux. Могу ли я подключиться к Интернету?

Питер Барнс [Peter Barnes]

О Ответ на ваш вопрос «да», но я подозреваю, что вы также хотите знать, как. У вас последовательный USB-модем: хотя для соединения с Интернетом он использует беспроводную сеть, в компьютере он выглядит как обычный телефонный модем. Заставить его работать в Linux можно с помощью весьма простых шагов. Первый – скачать драйвер и обеспечить его загрузку при каждом запуске. Второй – установить ПО для соединения. Модем нуждается в последовательном USB-драйвере, который присутствует в каждом дистрибутиве, но требует настройки для работы с вашим модемом, и для этого вам нужен его ID и ID изготовителя. Вы можете найти их, запустив *lsusb* в терминале от имени root или исследовав вывод *dmesg* (сообщения ядра). Откройте root-терминал, подключите модем и запустите

```
dmesg | tail -n 20
```

чтобы увидеть последние двадцать строк сообщений ядра и найти нечто вроде этого:

```
usb 2-4.4: new full speed USB device using ehci_cd and address 8
usb 2-4.4: new device found, idVendor=16d8, idProduct=6280
```

Теперь, используя значения из *dmesg* (они могут отличаться от приведенных здесь), загрузите драйвер:

```
modprobe -v usbserial vendor=0x16d8 product=0x6280
```

› Keeping a laptop and desktop in sync with Unison. »

! Вопрос-победитель (английская версия)

Кардридеры

В Я думаю купить камеру Canon S5IS, она использует SD-карты. Но мне сказали, что мне нужен еще и «кардридер». На моем компьютере достаточно свободных разъемов USB 2.0, но я не знаю, какой «ридер» совместим с Linux (у меня стоит Open SUSE 10.3).

Changeling, с форумов

О Гораздо сложнее ответить на вопрос «какой карт-ридер не работает в Linux?»

Некоторые люди сообщают о проблемах с кардридерами типа 99-в-одном – обычно это связано с запуском на самопальных ядрах, собранных без избыточных опций. Эти устройства часто нуждаются во включенной опции **SCSI_MULTI_LUN**, по умолчанию и так установленной в ядрах дистрибутивов, поскольку каждый порт карты представляется отдельным устройством. Если вы захотите использовать только один вид карт, более удобен одноформатный кардридер: он выглядит как USB-брелок, и вы просто вставляете в него карту и подключаете к USB-порту.

Вообще-то, кардридер вам может и не понадобиться: *GPhoto2* позволяет загружать изображения с большинства камер с помощью их USB-кабелей. вашей камеры нет в списке поддерживаемых устройств, но так как это новое устройство, оно может появиться позднее. Если Вы используете KDE, наберите **camera:/** в строке адреса *Konqueror*, и увидите список присоединенных камер и фотографий на них. Однако я считаю, что кардридер быстрее и удобнее, особенно если мне надо засунуть карту в мой ноутбук и продолжать снимать на другую. **НБ**

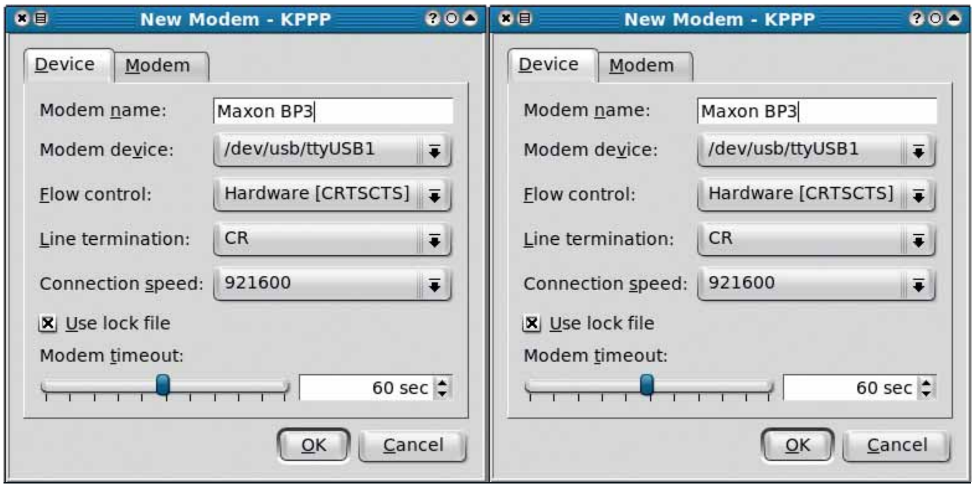
» Повторный запуск *dmesg* должен показать

```
usbcore: registered new interface driver usbserial
drivers/usb/serial/usb-serial.c: USB Serial support
registered for generic
usbserial_generic 2-4.4:1.0: generic converter
detected
usb 2-4.4: generic converter now attached to
ttyUSB0
usbserial_generic 2-4.4:1.1: generic converter
detected
usb 2-4.4: generic converter now attached to
ttyUSB1
usbserial_generic 2-4.4:1.2: generic converter
detected
usb 2-4.4: generic converter now attached to
ttyUSB2
```

и это означает, что вам удалось благополучно загрузить драйвер и активировать модем. Вы можете обеспечить это при загрузке, добавив **usbserial** в **/etc/modules.preload** и следующую строку – в **/etc/modprobe.conf**:

```
options usbserial vendor=0x16d8 product=0x6280
```

Теперь переключитесь в режим обычного пользователя. Как Вы можете видеть в выводе *dmseg*, модем представлен как три устройства. То, которое Вам нужно – **/dev/ttyUSB1**, поэтому запускайте *KPPP* и настраивайте новое соединение, используя **/dev/ttyUSB1** как устройство-модем, "***99#**" как номер для дозвона и прочие параметры подключения, предоставленные вашим провайдером. Настроив модем, установите Flow Control на Hardware и выключите Wait for dial tone before dialling. **НБ**



› Беспроводной широкополосный модем Maxon настраивается в *KPPP* как обычный телефонный модем.

6 ИнтерНЕТ или ИнтерДА?

ВЯ испробовал несколько дистрибутивов, включая Fedora, Mandriva и SUSE, и везде имел две проблемы. Первая заключалась в том, что *Firefox* постоянно терял связь с Интернетом, выдавая сообщение 'timed out' каждый раз, когда я пытался его обновить, хотя с *Konqueror* не было никаких проблем.

Вторая проблема в том, что иногда после нескольких минут, а иногда через час или два экран застывал и не реагировал на указатель мыши, который продолжал двигаться. Я использую видеокарту Nvidia, но безуспешно пытался установить проприетарные драйвера для нее. Попробовал Ubuntu Feisty с DVD LXF94, прочитав, что там есть простой метод установки драйверов Nvidia; сработало очень легко, и теперь мой экран перестал надолго застывать. Я также обнаружил, что *Firefox* работает в Feisty, как и обновления. Затем я решил попробовать Ubuntu Gutsy, когда он появился на DVD LXF100/101. Оказалось, что драйвера Nvidia не устанавливаются, и подключиться к Интернету тоже нельзя, и обновления не работают. Проблемы *Firefox* вернулись, а *Konqueror* продолжал работать.

Это проблема в моем компьютере или в настройках, и почему тогда работал Ubuntu Feisty?

Дэвид Сайкс [David Sykes]

ОПроблемы с интернет-соединением почти наверняка возникали из-за того, что ваш браузер пытался общаться с модемом или маршрутизатором через IPv6. Если соединение не работает через IPv6, то система предполагает возврат к старому IPv4. Однако некоторые маршрутизаторы не делают этого и сбиваются, когда клиент использует IPv6, а исходящее соединение (ваш провайдер) – нет.

Есть три решения: исправить маршрутизатор, выключить IPv6 на системном уровне или выключить его в браузере. Будет полезно проверить сайт изготовителя маршрутизатора и посмотреть на нем наличие обновлений прошивки. После перепрошивки проблема может полностью исчезнуть.

Часто задаваемые вопросы...

Rsync и Unison

Синхронизация двух директорий без потери файлов.

» Что такое *Rsync*?
Это способ синхронизации содержимого двух директорий, причем обе будут полностью идентичны.

» А что, нельзя использовать для этого cp?
cp копирует все подряд. *Rsync* – только те файлы, которые различаются. Если изменились большие файлы, копируются только измененные части.

» Неплохо. А как ее использовать?

```
rsync --archive --delete /path/to/
source/ /path/to/dest/
```

точно скопирует первую директорию во вторую. Параметр **--delete** удаляет файлы, которых нет в первой директории, а **--archive** копирует также права доступа и временные отметки. Замыкающие слэши важны для *Rsync*: они указывают, что вы хотите синхронизировать содержимое директорий. Без слэшей программа просто скопирует одну директорию в другую.

» Можно ли синхронизировать директории разных компьютеров? У меня локальная копия web-сайта, и не хотелось бы копировать все подряд, когда изменится только пара файлов.
При наличии SSH-доступа к web-серверу, можно сделать так:

```
rsync --archive --delete /my/local/site
server:/path/to/site/
```

Если один из путей или оба содержат имя хоста, *Rsync* использует для передачи удаленную оболочку. По умолчанию это SSH, но можно заменить ее любой другой – даже RSH, если вы не против, чтобы ваши файлы во время передачи читал кто ни попадя!

» А как синхронизировать директории между Linux и Windows-компьютерами?
Rsync работает и под Windows, но параметр **--archive** на разделах FAT действовать не будет, ведь FAT не поддерживает права доступа. Подробности можно узнать в man-странице программы.

» Все это как-то в одни ворота.
Тогда вам нужна программа вроде *Unison* (www.cis.upenn.edu/~bcpierce/unison), специально задуманная для этой задачи. Для синхронизации *Unison* пользуется той же *Rsync*, а для передачи – SSH, но она предназначена для двусторонней синхронизации. Если на двух компьютерах изменились некоторые файлы, каждый компьютер получит новейшую версию.

» Умно придумано; а если я отредактирую один и тот же файл на обоих компьютерах?
Программа не умеет читать мысли и решать, какая версия вам дороже, она просто сообщит, что обе копии редактировались, и спросит, какую из них сохранить. *Unison* отслеживает все изменения, и заметит, что файл был изменен на обоих компьютерах.

Если нет, можно велеть вашему компьютеру не использовать IPv6, чтобы *Firefox* или любая другая программа даже не пыталась соединиться с маршрутизатором таким образом. Выключите IPv6, добавив или отредактировав следующие две строки в файле конфигурации модулей, а именно **/etc/modprobe.d/aliases** в Ubuntu, или, в других дистрибутивах, **/etc/modprobe.conf** или **/etc/modules.d/aliases**.

```
alias net-pf-10 off
alias ipv6 off
```

Они должны заместить любые имеющиеся строки со ссылкой на IPv6. Третий способ, который нужно применять, если ничего не помогло, это выключить IPv6 в *Firefox*, но он не поможет ничему кроме браузера. Наберите **about:config** в строке адреса *Firefox*, затем IPv6 в поле Filter. Если **network.dns.disableIPv6** установлена в False, кликните на нее правой кнопкой и отметьте пункт в меню для смены на True.

Большинство дистрибутивов теперь имеют драйвера Nvidia в своих репозиториях, но их установка также проста, если скачать пакет с ними с **www.nvidia.com**. Нажмите Ctrl+Alt+F1 для переключения с графического окружения в консоль, зайдите как root (или запустите *sudo -i* в Ubuntu), затем выполните команду

```
init 3
sh NVIDIA-Linux-XXXX-pkg1.run
nvidia-config
init 5
```

Пользователи Debian и Ubuntu должны заменить первую и последнюю команду, выключающую и включающую *X*-сервер, на

```
/etc/init.d/dm stop
/etc/init.d/dm start
```

Установщик Nvidia не запустится, если увидит *X*-ы, отсюда необходимость переключения в консоль и их останова. Часть инсталляционного процесса может подразумевать компилирование модуля для Вашего ядра, и если произойдет сбой, убедитесь, что у Вас установлен пакет *build-essentials*. *Nvidia-config* модифицирует файл настроек *X* для использования нового драйвера. **НБ**

7 Просто вставь в розетку

В У меня есть dLAN HomePlug 85Mbps Ethernet Starter Kit от Devolo. Devolo предоставляет программную утилиту для включения шифрования между устройствами, но я не знаю, как установить эту программу.

Скотт [Scott]

О Это пакет исходных текстов: его нужно скомпилировать, предварительно установив несколько требуемых зависимостей с помощью *Synaptic*. *Dlanconfig* зависит от *libpcap*, который по умолчанию уже установлен в Ubuntu, но для новой компиляции необходим пакет *libpcap-dev*. Вам также понадобится пакет *build-essentials*, включающий компилятор и другие инструменты, необходимые для сборки ПО из исходников. Установив их через *Synaptic*, откройте терминал и зайдите в директорию, содержащую скачанный с **www.devolo.com** файл. Затем запустите

```
tar xf dLAN-linux-package-XX.tar.gz
cd dLAN-linux-package-XX
./configure
make cfgtool
sudo make install-cfgtool
```

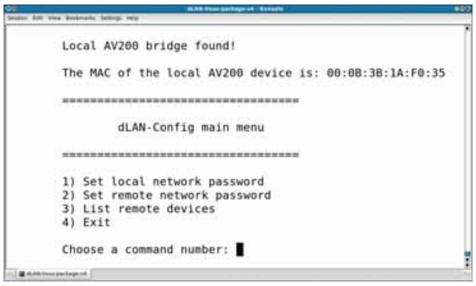
Первые две строки распаковывают архив и меняют директорию на содержащую полученные в результате файлы. Следующие две собирают инструмент настройки из исходных кодов, а последняя строка устанавливает его в систему. Если вы вдруг захотите удалить это ПО, повторите процесс, заменив последнюю команду на

```
sudo make uninstall
```

Теперь вы можете запустить программу с помощью

```
sudo dlanconfig eth0
```

и сменить свои пароли. вам может понадобиться секретный код устройства, который нельзя прочесть, когда оно подключено, поэтому позаботьтесь об этом заранее. Запустите *Dlanconfig* и выберите опцию смены удаленного пароля, при этом сменится также и пароль локального устройства. Если Вы сперва смените его на локальном устройстве, то не сможете потом подключиться к удаленному устройству, так как пароли не совпадут.



› **Установите программное обеспечение Devolo dLAN для смены настроек шифрования вашей домашней сети HomePlug.**

Если у вас USB-версия устройства dLAN, надо будет изменить последние две строчки процесса сборки на

```
make usbdriver
sudo make install-usbdriver
```

или просто

```
make
make install
```

чтобы инсталлировать обе версии. Собранный USB-драйвер требует наличия исходных текстов ядра, поэтому сперва используйте *Synaptic*, чтобы установить пакет linux-source. **НБ**

8 Файлы для всех даром

В Недавно я купил новый компьютер и поставил на него Xubuntu 7.10. На старом ПК использовалась Windows и было заведено два пользователя (для меня и моей супруги). Теперь стоит задача переноса всех документов и мультимедийных файлов. Мы использовали их совместно под разными учетными записями и хранили в папке «Общие документы», к которой имели доступ оба пользователя. Как организовать общую папку в GNU/Linux, чтобы каждый пользователь системы мог иметь полный доступ к ней? По умолчанию, каждый пользователь «видит» только свою домашнюю папку, но не домашние папки других пользователей. Или, возможно, есть другие варианты организации совместного доступа к файлам?

Сергей Винаков

»

Краткая справка про...

Команды

При работе в командной строке часто возникает необходимость последовательного запуска нескольких команд (это также может произойти, если Вы используете графический интерфейс, однако там решение не так просто). Классический пример – компиляция ПО из исходных текстов, когда запускается *./configure* (возможно, с аргументами), а потом *make* и *make install*.

Каждый из этих последовательных шагов занимает от нескольких минут до нескольких часов, в зависимости от сложности кода и мощности машины. Дожидаться окончания одного, чтобы перейти к другому, неэффективно, поэтому можно поступить так:

```
./configure; make; make install
```

Точка с запятой вызывает последовательное выполнение команд, как будто их запустили по очереди.

Вы наверняка углядели потенциальную проблему: если *./configure* или *make* дадут сбой, последующие команды все равно выполнятся? Ответ – да: сообщения об ошибках будут скрыты, и Вы ничего даже не заподозрите, пока не запустите программу.

Безопаснее связывать команды так:

```
./configure && make && make install
```

&& – это логический оператор. Строка фактически означает следующее: «если *./configure* истинно и *make* истинно и *make install* истинно». На наше счастье, оболочка определяет истинность команды ее запуском и наблюдением за ошибками. В случае отказа команды необходимость в запуске следующей отпадает, так как тест уже провален, поэтому объединение команд через **&&** запускает их последовательно, но прекращает работу при первом же сообщении об ошибке (а значит, больше незачем караулить оболочку).

Компаньон команды **&&** – это ||, что значит ‘or’ (или). Так, в выражении

```
command1 || command2
```

command2 работает только в случае провала **command1**. В интерактивной оболочке это особо ни к чему, но часто используется в скриптах:

```
somecommand1 || echo "Что-то рухнуло!"
```

О Я думаю, в вашем случае проще всего поступить, как раньше. То есть:

» Создать от имени root папку, скажем, **/home/shared** (можете использовать и русское “Общие документы”, но я по старой привычке не люблю называть файлы и каталоги кириллицей):

```
sudo mkdir /home/shared
```

» Установить ей подходящих владельца и группу владельца (например, **root:users**, важно лишь, чтобы группой владельца была **users** – т.е. та, к которой принадлежат ваши с женой пользователи)

```
sudo chown root:users /home/shared
```

» Установить права доступа: все для владельца, все для группы и ничего – для остальных:

```
sudo chmod 0770 /home/shared
```

» Необязательный шаг: установите на каталог т.н. «sticky bit», чтобы писать в него файлы могли все, но каждый мог удалять лишь то, что принадлежит ему:

```
sudo chmod +t /home/shared
```

Последние две команды можно объединить (скажем, так: *sudo chmod ug+rwxt,o-rwxt /home/shared*) – я разнес их только для того, чтобы сделать четвертый шаг опциональным.

После того, как общие файлы будут скопированы, не забудьте установить для них аналогичные права доступа командой *chown/chmod -r*. **ВС**

9 Кина не будет?

В Иногда я люблю заходить на новостные сайты и смотреть там видеоролики, например, http://news.ntv.ru/133377/ или http://www.russiatoday.ru/news/news/25473/video. Поэтому для меня важно, чтобы все это работало и в Linux, на который я подумываю перейти с Windows XP. Однако, когда я экспериментировал с этой ОС, у меня не получалось посмотреть видео на некоторых интернет-страницах, до сих пор не могу понять, почему... а такой сервис, как *YouTube* – работал.

Все-таки интересно, в чем здесь может быть дело, подскажите?

Александр Тасаев

О Для проигрывания потокового видео с сайтов в браузере нужно поставить плагин (англ. plugin). Это независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования ее возможностей.

Первый плагин – *FlashPlayer*, скорее всего, у Вас уже стоит, поскольку для просмотра видео на *YouTube* требуется именно он. Время от времени его необходимо обновлять, поскольку многие сайты довольно быстро переориентируются на новые версии *FlashPlayer*.

Второй необходимый плагин – это *mplayerplug-in*, предназначенный для проигрывания потокового видео *Windows Media* (файлы с расширениями **.wmv**, **.avi**, **.asf**, **.wav** и **.asx**), *QuickTime* (**.mov** и **.smil**), MPEG Video and Audio (**.mpeg** и **.mp3**), *Ogg Vorbis* (**.ogg**), *AutoDesk FLI* (**.fli** и **.flc**), *Vivo* (**.vivo**) и *Real Player* (**.ram** и **.rm**). Его установка отличается от дистрибутива к дистрибутиву, но общие инструкции, вкратце, таковы. Во-первых, необходим сам *mplayer*, иначе некоторые форматы кодеков не будут работать. Во-вторых, требуется удалить плагины других видеопроигрывателей, таких как *Totem*, *VLC*, *Kaffeine* и т.п., если они у вас установлены. В разных дистрибутивах они могут называться по-разному: в *Mandfriva*, например, это *mozilla-plugin-vlc* и *totem-mozilla*. Затем нужно установить сам mplayerplugin, предпочтительнее всего – средствами пакетного менеджера вашего дистрибутива. В большинстве случаев при инсталляции будет добавлена поддержка практически всех видеоформатов. Более подробные инструкции можно найти по ссылке: http://mplayerplug-in.sourceforge.net/.

Проверить, какие плагины установлены в вашем браузере, можно, набрав в адресной строке команду **about:plugins**. **ЮД**

Нужна помощь!

› Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

› Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Мы стараемся ответить на все вопросы. Если вы не нашли ответ на свой, это, возможно, потому, что мы уже ответили на похожий вопрос.



Большой вопрос Как синхронизировать мой iPod с Linux?

10 Использование iPod с Linux

В Только что приобрел iPod nano – и уже наткнулся на проблемы при использовании его с Linux. При каждой попытке записать на него музыку он выдает ошибку. *Banshee* и *Gtkpod* забивают все место в моем iPod, ничего на него не помещая, и он не монтируется в *Amarok*. Это очень досадно, так как я вынужден записывать мои песни на iPod с семейного компьютера, где установлена Windows. На моем компьютере стоит Ubuntu 7.10.

Джош Ансел [Josh Ansell]

О Есть несколько способов передачи музыки между Linux и iPod. *Banshee* и *Amarok* обе включают эту функцию, а *Gtkpod* – специальная программа для данной задачи. Есть и способы интеграции iPod с другим ПО, такие как файловая система fusepod от Fuse и *Kpod* KIO slave для KDE. Хотя возможно смонтировать iPod просто как устройство хранения USB, обычное копирование на него файлов ничего не делает, лишь заполняет его. Вам нужен соответствующий инструмент для управления музыкой на устройстве от Apple.

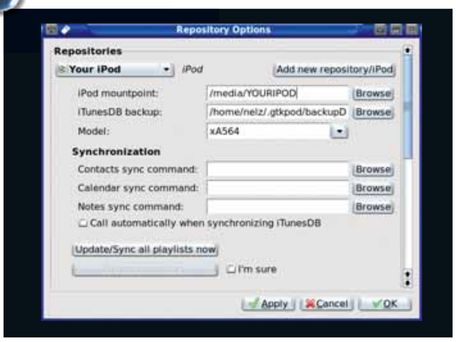
Самое простое – попробовать *Gtkpod*. Запустите программу и зайдите в Edit > Edit Repository/iPod Options. Установите точку монтирования, куда автоматически подключается iPod, и выберите правильный номер модели. Вы можете найти его на устройстве в первой строке файла **iPod_Control/Device/SysInfo**. Теперь вы можете добавить музыку в коллекцию *Gtkpod* и скопировать треки на iPod перетаскиванием их из локального окна внутрь iPod при нажатой левой кнопке мыши. Затем надо кликнуть Save Changes для записи файлов на iPod и, что очень важно, обновить базу данных iPod.

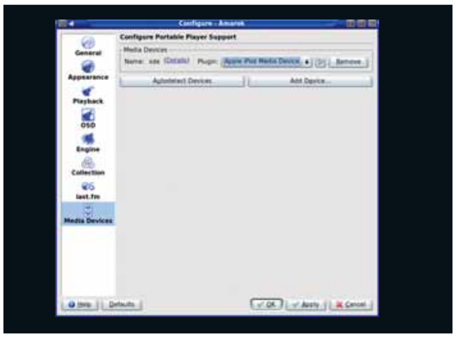
Процедура похожа и при использовании *Amarok*. Сперва скажите ему о вашем устройстве в основном окне настройки, затем зайдите во вкладку устройств в главном окне и кликните в нем на иконку iPod для выбора правильной модели. Теперь можно перетаскивать треки или целые плейлисты на ваш iPod. Они будут появляться на странице передачи слева внизу окна, пока вы не нажмете кнопку Transfer. Этот шаг необходим для копирования треков и обновления базы данных. Возможно, необновленная база и вызывает ваши проблемы, что бывает при неверном выборе модели iPod.

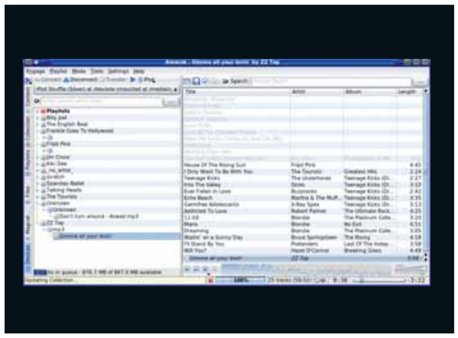
Если вы предпочитаете *Banshee*, то он тоже копирует песни на iPod, и его проще всех настроить. Просто скажите, где живут ваши музыкальные файлы, подключите iPod и перетащите треки из библиотеки. Скопировать файлы и обновить базу iPod можно, щелкнув правой кнопкой на имени iPod в левой панели и выбрав Synchronise iPod и Save Manual Changes. Как альтернативу, можете использовать опцию Synchronise Library для копирования вашей коллекции на iPod.

Для доступа к устройству напрямую существует файловая система Fuse для iPod, доступная на http://fusepod.sourceforge.net, и KDE KIO slave с http://sourceforge.net/projects/Kpod. В последнем случае Вы можете набрать **ipod:/** в адресной строке *Konqueror* для прямого доступа к iPod и скопировать файлы, перенеся их в папку **Transfer**. Затем зайдите в папку **Utilities** и запустите Synchronise для записи изменений. **НБ**

Шаг за шагом: Инструменты настройки iPod







1 Настройка Gtkpod

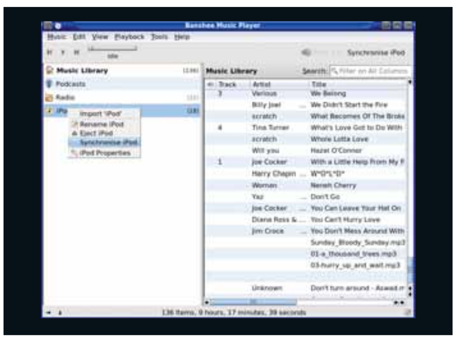
Gtkpod легко настроить, но вам может понадобиться заглянуть в файловую систему iPod для поиска корректного номера модели.

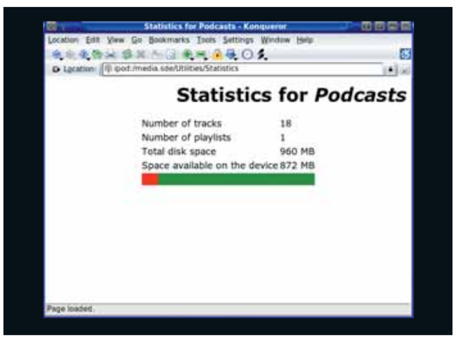
2 Настройка Amarok

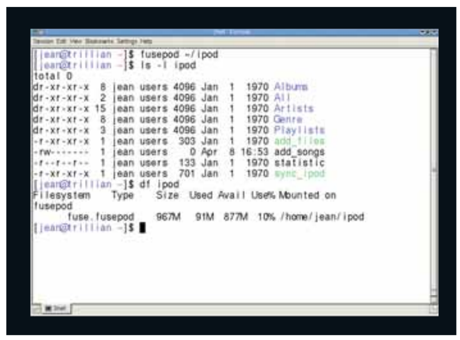
Amarok может и сам определить ваш iPod: проверьте детали и подтвердите их. Установите модель из вкладки Devices главного окна.

3 Использование Amarok

Определив iPod, можете проигрывать музыку с него или копировать треки, но убедитесь, что после копирования выполнили синхронизацию.







4 Banshee

Интерфейс *Banshee* лаконичный и простой в использовании. Перетащите треки для копирования и правым кликом на iPod синхронизируйте их.

5 iPod как устройство в KDE

Kpod KIO slave предоставляет прямой доступ на чтение и запись музыки на iPod и включает множество утилит.

6 iPod в консоли

Файловая система fusepod позволяет монтировать iPod и работать с его базой в командной строке – интерфейс не в духе Apple, но он работает.



Лучшие новинки открытого ПО на планете

LXF HotPicks



Ричард Смедли
Активист сообщества FOSS и давний сотрудник **LXF**, Ричард рыщет по лесу в сопровождении ученой обезьяны, вынюхивая трюфели HotPick.

В ЭТОТ РАЗ ТОЛЬКО ДЛЯ ВАС: Chandler » SoundConverter » Super Grub Disk » Xmonad » Risk » QTads » Rootz » PictureFlow » Wammu » WordPress

Менеджер личной информации

Chandler

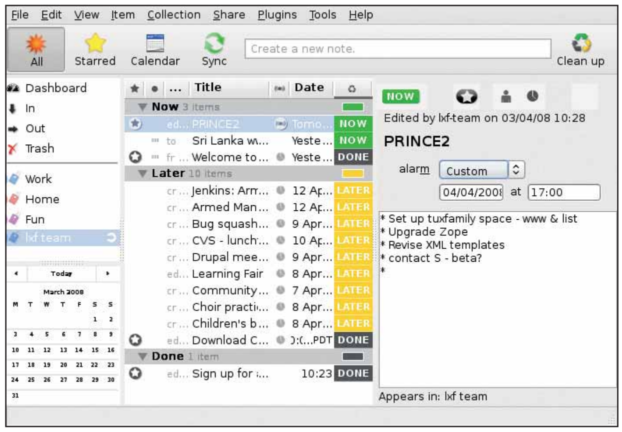
Версия 0.7.5 **Сайт** http://chandlerproject.org

У вас много работы? Тогда вы наверняка тратите немалую часть мозговых усилий на то, чтобы просто упомнить все свои дела, а вовсе не на то, чтобы собраться и сделать их. Каждое новое задание, полученное вами от менеджера/преподавателя/иного руководителя, снижает вашу способность справиться с ним. По крайней мере, мы всегда оправдываемся именно этим.

Дэвид Аллен [David Allen] несколько продвинулся по пути к решению этой головоломки, создав методику управления временем «Как привести дела в порядок» ('Getting Things Done'): она гласит, что нужно ассоциировать каждую задачу с определенным действием, потом записать, что вы должны сделать, и вернуться к этому позднее. Авторы *Chandler* зашли еще на шаг дальше и применили теории Аллена к менеджеру личной информации, созданному для перегруженных делами работников интеллектуальной сферы.

Управление знаниями

Удобнее всего рассматривать *Chandler* в качестве продвинутого списка задач. Данные вводятся (или импортируются) в *Chandler* как заметки – их можно добавлять в календарь в виде событий, отправлять по электронной почте другим или выделять «звездой». Заметки объединяются в группы, отмеченные ярлычками «Сейчас» ('Now'), «Потом» ('Later') и «Готово» ('Done') в соответствии со срочностью их выполнения; причем события, занесенные в календарь, автоматически сортируются в соответствии с тем, когда они были выполнены. Количество групп неограниченно, и стоит вам привыкнуть организовывать свою работу таким образом, вы обнаружите, что это стало вашей второй натурой.



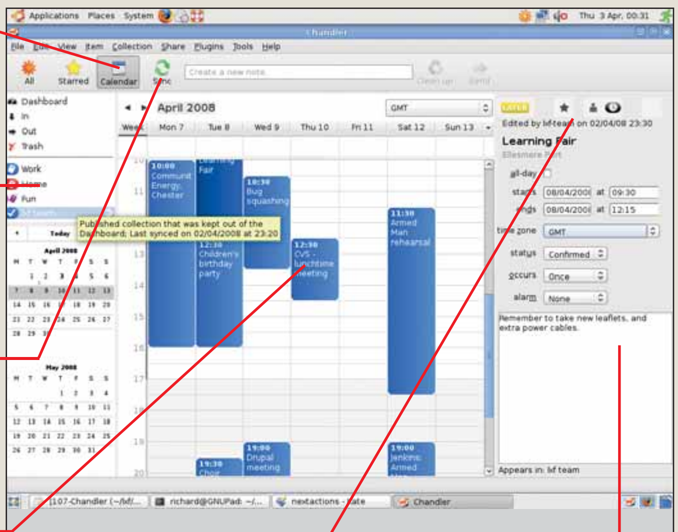
› В *Chandler* имеются все функции обычного менеджера личной информации, например, заметки, связанные с сигналами.

Chandler существует не только в виде настольной программы: разработчики создали также и *Chandler Hub*, web-приложение, позволяющее пользователям *Chandler* делиться информацией. Ваша учетная запись в *Chandler Hub* предназначена для совместной работы в малых группах, в отличие от *Evolution* или *Outlook*, предназначенных для более солидных корпоративных контактов.

На сайте *Chandler* – особенно в блоге проекта – немало документации, отражающей усилия, позволившие *Chandler* достичь таких успехов, но информация довольно путаная, и в ней бывает трудно разобраться. Если вы хотите лучше понять принципы работы *Chandler*, отправляйтесь в ближайшую библиотеку, и возьмите там книгу Дэвида Аллена *«Как привести дела в порядок. Искусство продуктивности без стресса»*.

Большая часть истории создания *Chandler* описана в книге Скотта Розенберга [Scott Rosenberg] *«Мечтающий в коде: две дюжины программистов, три года, 4732 ошибки и один поиск Идеальной программы»* [*Dreaming in Code: Two Dozen Programmers, Three Years, 4,732 Bugs, and One Quest for Transcendent Software*], опубликованной в прошлом году и подробно разбирающей проблемы, присущие любому крупному проекту. Прочитав эти две книги, берите *Chandler*, станьте организованными и тратьте больше времени на работу, а меньше – на всякие переживания!

Исследуем интерфейс Chandler



Просмотр
Переключайтесь с календаря на общий вид; включение фильтра ограничит просмотр только «звездными» событиями.

Работа и игра
Пусть ваши дела и ваши хобби хранятся в одном приложении – тогда вы увидите соотношение работы и личной жизни.

Синхронизация
Открывает доступ к серверу *Chandler*, чтобы проявить всю мощь приложения для организации совместной работы.

Календарь
Ввод времен в календарь с помощью мыши – одно из удовольствий *Chandler* (просмотр задач показан на другом экранном снимке).

Интерфейс одним щелчком
Изменение статуса сортировки, добавление/удаление «звезды», отправка электронной почтой и добавление в календарь.

Добавление заметок
Создает заметки и сигналы для присвоения календарным событиям.

Конвертор аудиофайлов

SoundConverter

Версия 1.0.0 **Сайт** http://soundconverter.berlios.de

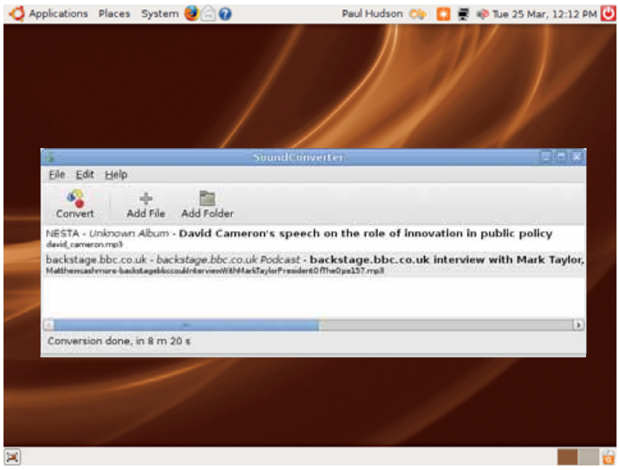
Порою кажется, что мы завалены MP3-плейерами, и что скоро они будут бесплатно прилагаться к пачкам каши для завтрака. Средства для прослушивания аудио есть в телефонах, автомобилях, чуть ли не в каждом бытовом устройстве. Программы вроде *Banshee* и *Amarok* позволяют обновлять большие списки воспроизведения для iPod'ов и других плейеров, но чтобы просто добавить пару файлов в телефон для прослушивания по дороге на работу, они все-таки слишком роскошны.

Обратитесь к *SoundConverter*, приложению Gnome для создания OGG, MP3 или сжатых без потерь файлов из практически любого формата. Благодаря движку *GStreamer* и разнообразным модулям расширения, он идеален для создания маленьких файлов для портативных устройств без внесения изменений в вашу основную коллекцию.

Улучшения

SoundConverter создан, чтобы выполнять одну задачу просто и хорошо. Множество мелких улучшений и исправлений ошибок привели к выходу релиза 1.0; хотя разработчики и подвергались искушению добавить в него ненужных функций, но устояли. Первым делом зайдите в Preferences [Настройки] в меню Edit [Правка], чтобы задать параметры по умолчанию. Выберите файл для размещения полученного результата и, если захотите, подпапки для исполнителя или альбома. Можно также настроить изменения в имени файла и сообщить приложению, надо ли удалять исходные композиции. И, наконец, задайте формат конечного результата – WAV, FLAC, MP3 или OGG, и качество (битрейт), если вы используете формат с потерями.

> «Banshee и Amarok — это роскошь, если надо просто добавить музыку в телефон.»



› **Найдите нужный вам файл и кликните на Convert – чего уж проще? Можно даже создать MP3 и OGG из MOV и AVI.**

Теперь остается только нажать на кнопку Convert. Строка состояния покажет вам время, необходимое для конвертирования, и вы сможете приостановить процесс, если потребуется высвободить ресурсы для выполнения другой задачи. Любой современный дистрибутив с *GStreamer* и библиотеками Gnome позволит запустить *SoundConverter*, при наличии библиотек *gnome-python* и *PyGTK*.

Спасательный диск

Super Grub Disk

Версия 0.9711 **Сайт** http://supergrubdisk.org

Команда **LXF** вечно все устанавливает, переустанавливает и обновляет, и вроде бы должна уже лихо разбираться в командах *Grub*, но ошибки все равно случаются. Дистрибутивы для восстановления системы типа Insert (см. LXF105), конечно, хороши, но бывают проблемы, не решаемые без специалиста. В таком случае – да здравствует *Super Grub Disk* (*SGD*): это загружаемая дискета или CD-ROM, позволяющие выбрать в меню любой из имеющихся сценариев восстановления системы, например, перезапись загрузчика, затертого переустановкой Windows. Вы можете восстановить *Grub* в MBR и загрузиться с любого накопителя, подключенного к компьютеру, а этот новый релиз прямо чудеса творит с разделами Windows на других дисках. Если данных опций вам недостаточно, вы всегда можете нажать C и перейти в командную строку *Grub*.

Архив с *SGD* – крошечный, размером всего лишь 400 КБ в *bzip* (если надо его загрузить из ОС, не поддерживающей декомпрессию файлов bzip2, имеется ISO-файл размером 4 МБ). Есть также образы для дискет и USB-брелков. Для записи файлов на «флешку», отформатируйте раздел (как ext2 или ext3), затем используйте *Grub*, чтобы сделать ее загружаемой. Для этого понадобится компьютер с Linux и с установленным *Grub*. Скопируйте извлеченные из архива файлы в **/boot** на отформатированном брелке, затем размонтируйте накопитель и запустите *Grub* из терминала:

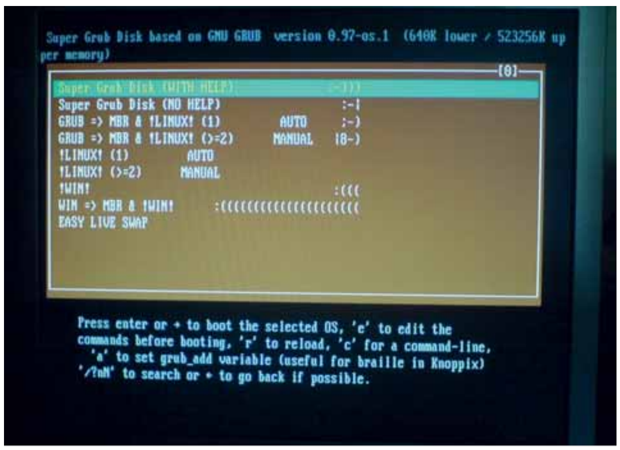
```
grub
grub>device (hd3) /dev/sdc
grub>root (hd3,0)
grub>setup (hd3)
grub>quit
```

где **(hd3)** и **/dev/sdc** изменены должным образом: например **(hd2,0)**.

Шикарная загрузка

Загрузитесь с любого носителя, и перед вами появится меню, где самым популярным выбором будет первый ('with help'). Затем, предоставляется огромный выбор из большинства европейских языков, в том числе тех, на которых говорят в Испании – что выдает родину проекта: Иберийский полуостров. После этого выбираются необходимые опции восстановления системы. *SGD* загрузит не только разделы, но и слайсы (дисковые сегменты, используемые BSD и OpenSolaris), так что вы сможете восстановить свою систему после всех экспериментов с операционкой.

На сайте *SGD* имеется достаточно информации по большинству сценариев, а также ссылки на полезные документы, в том числе и на расширенное руководство Шона Хермана [Shawn Herman] по сборке и использованию *SGD*.



› **Ну, *Grub*, давай! Вздохните с облегчением, когда незагружаемое вновь станет загружаемым.**

Оконный менеджер

Xmonad

Версия 0.7 **Сайт** http://xmonad.org



› **Освободите себе жизненное пространство с помощью эффективного мозаичного менеджера окон, подобного *Xmonad*.**

Недавно разорились на большой LCD-монитор? Ну, скоро вы увидите, что ваши трудовые денежки вложены в бесполезную трату экранного пространства на рамки окон. А уж если вы приобрели одно из новых мини-чудес типа Eee PC, вы еще меньше будете склонны тратить драгоценные пиксели на оконные декорации.

Существует целая школа минималистских менеджеров окон, цель которой – высвободить место на экране и сэкономить вам ненужные перемещения мыши, и некоторые из ее представителей освещались в нашем Сравнении в LXF103. *Xmonad* – новичок отряда эффективных GUI, стремящихся к минимизации размера, и его недавняя разработка шла очень резво. *Xmonad* написан на Haskell – этот язык программирования используется также в системе управления версиями *Darcs*. Для Ubuntu и некоторых дистрибутивов имеются пакеты, а для иных Linux'ов и BSD – готовые двоичные файлы, но сборка из исходников не вызовет затруднений. Возьмите самый свежий tar-архив с сайта *Xmonad* или скачайте снимок кода из репозитория *Darcs*. Вам понадобится *Glasgow Haskell Compiler* (в большинстве систем он именуется *GHC6*) и заголовки X11 (которые обычно находятся в пакете под названием **libx11-dev**, если они еще не установлены).

Под вашими пальцами

Работать в *Xmonad* легко, но понадобится посвятить некоторое время изучению клавиатурных команд. Alt+Shift+Enter открывает терминал; если повторить эту команду, она опять откроет терминал, который выстроится в мозаику с первым. Alt+Space переключает между мозаикой и полноэкранным отображением выделенного окна. Alt+J и Alt+K передает фокус с одной панели на другую, как и мышь. Другие комбинации перемещают выделенное окно на разные панели и изменяют размеры отдельных панелей. Щелчок и удержание кнопки мыши сделает окно плавающим, а Alt+T вернет его на прежнее место в мозаике.

И много чего еще...

К мощи готового *Xmonad* можно много чего добавить. *Dmenu* даст вам доступ к любому приложению вашей системы, и его легко загрузить, распаковать и установить. Как только оно появится в системе, нажмите Alt+P, и откроется меню в виде строки с названиями программ в верхней части экрана. Начните набирать название программы, которую вы хотите запустить, и перечень в этой строке будет уменьшаться, пока не останется единственная опция; можно и просто выбрать приложения из меню.

Пакет **xmonad-contrib** содержит добавочные «крючки» [hooks] и расширения, меняющие алгоритмы раскладки и мозаики окон или рендеринг шрифтов, а также различные комбинации для приглашений ввода, *SSH* и прочего. Написание или модификация модуля для пакета **xmonad-contrib** – отличный способ попрактиковаться в кодировании на Haskell. Haskell не похож на большинство распространенных языков, и позволит вам по-новому взглянуть на программирование.

Почти все менеджеры окон в Unix имеют множественные рабочие столы – явление, незнакомое большинству пользователей других ОС; и *Xmonad* – не исключение. По умолчанию здесь девять рабочих пространств, и они открываются с помощью комбинаций от Alt+1 до Alt+9. Если у вас более одного монитора, каждый из них может показывать свой рабочий стол, или же используйте *Xmonad* с *Xinerama*, чтобы устроить показ на стольких экранах, со сколькими справляется ваша видеокарта (как показано выше).

Этот самый свежий релиз может использоваться в качестве замены обычного менеджера окон в Gnome, Metacity. Как правило, для этого надо отредактировать **~/.gnomerc**, добавив

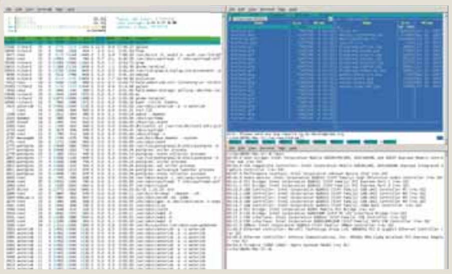
```
export WINDOW_MANAGER=xmonad
```

затем настроить *Xmonad*, чтобы освободить место для панели и строки состояния Gnome. Вы можете отключить *Nautilus* с помощью:

```
gconftool --type boolean --set /apps/nautilus/preferences/show_desktop false
```

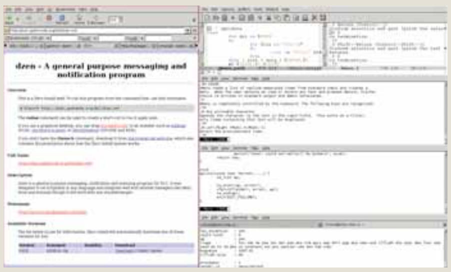
Другие необходимые настройки, в частности, изменения, которые надо сделать в **~/.Xmonad/Xmonad.hs**, можно найти на Haskell wiki, ссылка на которую имеется на сайте *Xmonad*.

Шаг за шагом: жизнь с мозаичными окнами



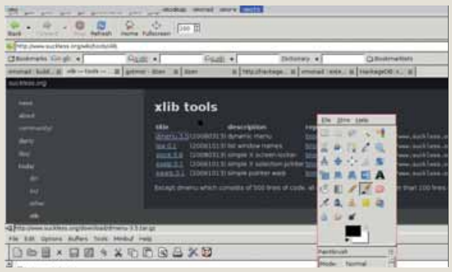
» Назад, в 80-е

При каждом нажатии Alt+Shift+Enter открывается новый терминал – в данном случае, Gnome Terminal – без излишних оконных украшений.



» Раскладка мозаики

Когда вы работаете на определенной панели, можете нажать Alt+Space, чтобы получить мозаичное расположение окон, какое вам нравится, а Alt+H и Alt+L растянет его до нужных размеров.



» Перемещение и меню

Комбинации от Alt+1 до Alt+9 перемещают вас по рабочим столам, подобно менеджеру окон *PWM*. Alt+Shift+3 переместит выделенную панель на третий рабочий стол.

HotGames Развлекательные приложения

Игра-стратегия

Risk

Версия 1.0.9.5 **Сайт** http://jrisk.sourceforge.net

Еще не познали вкус приключений – преодоления вулканов Камчатки, марша-броска через Урал и захвата всего на своем пути перед взятием Европы? Вы пропустили скромную кровавую баню, каковой является настольная игра *Risk*. Не теряйте время: скачайте это новое Java-издание и выпустите на волю псов войны.

Загрузив и распаковав эту программу, можете кликнуть на **Risk.sh** в своем файловом менеджере, или просто запустить **java -jar Risk.jar** из получившейся директории.

Вам понадобится достаточно мощный компьютер, но теоретически *Risk* будет работать на любой машине выше Pentium III с 64 МБ ОЗУ. Играть можно и на ноутбуке с экраном 1024x768, но этот, самый свежий релиз, позволяет устанавливать намного большие разрешения в версии Swing GUI. Можно сразиться с пятью ботами или включить режим многопользовательской игры. Любой компьютер, на котором запущен *Risk*, может служить игровым сервером – просто выберите Start Server [Запустить сервер] в меню. Другие игроки должны нажать на Join Game [Присоединиться] и ввести адрес сервера. Вы можете играть по локальной сети или через Интернет. Когда в игру войдут все человеческие игроки, для полноты ощущений можно добавить ботов.

Игроки по очереди выставляют свои армии на карту игрового мира, разделенного на территории. Каждый раз надо добавлять на несколько армий больше, и вы выбираете, какие соседние территории захватывать. Чем больше стран и территорий вы захватили, тем больше армий вы получаете для следующего хода. Результат битвы определяется подбрасыванием игрового кубика – это действие за вас любезно выполнит ваш компьютер.



› Краткий миг триумфа зеленых — и вот уже команды ботов начисто смели нас с арены.

Особой деликатностью стратегия *Risk* не грешит. Численное превосходство в нужное время дает возможность победить, так что будьте осторожны и внимательны, размещая свои армии – пусть другие игроки поубивают друг друга и выйдут из игры. В настольной игре часто создаются и разрушаются союзы – если вы играете с ботами, лучше никому не доверять. И помните – это всего лишь игра, а не метафора жизни.

Платформа интерактивной фантастики

QTads

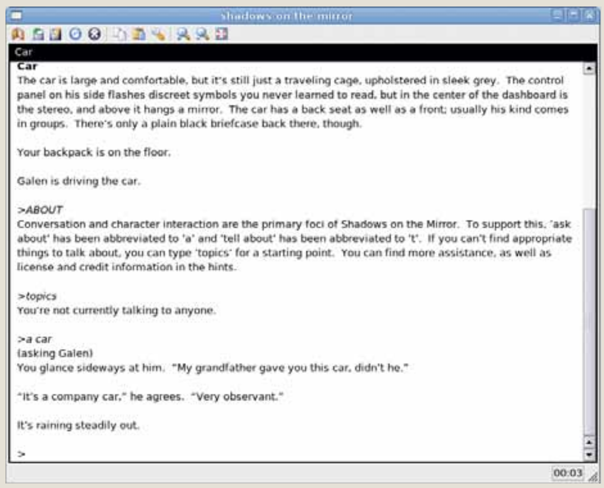
Версия 1.7 **Сайт** http://qtads.sourceforge.net

В далекие 8-битные дни, когда писатели ваяли свои романы на пишущей машинке Remington, текстовые приключения были основным видом игр, унаследованным еще от университетских мейнфреймов. Приключения вроде *Colossal Cave Adventure* («Вы – в лабиринте из одинаковых извилистых проходов») привели к появлению интерактивной компьютерной фантастики (interactive fiction, IF) типа *Zork* («Темно, хоть глаз выколи. И вас могут съесть») и подарили нам две из самых известных фраз, бытовавших в Интернете, пока не появилась «Вся ваша база есть принадлежит нам» [досл. "All your base are belong to us" – грамматически ошибочная фраза из англоязычной версии игры *Zero Wing*, – *прим. пер.*]. Взаимодействие с пользователем сводилось к вводу вопросов, а не к щелчкам мышью в лабиринте.

С тех пор текстовые приключения эволюционировали в современные графические, а IF выросла, превратившись в гипертекстовую фантастику и игровые книги (gamebooks). *Tads* (Text Adventure Development System, система разработки текстовых приключений) была shareware-приложением для разработки и запуска IF, а *Tads 2* стала в 1980-х кроссплатформенным стандартом для производства IF; затем последовала *Tads 3*, ее объектно-ориентированная редакция.

FrobTads – это интересная чисто консольная версия, с открытым кодом и возможностью свободного распространения. Увы, модифицировать или улучшать *Frobtads* не разрешается, но ее можно портировать, и она появилась на многих платформах. Единственный вариант для истых сторонников свободного ПО – *Qtads*. Он делает свое дело хорошо, а в новой версии имеется множество полезных обновлений.

Помимо компиляции для 64-битных машин, в *QTads 1.7* также увеличилось число откатов ходов; программа анализирует и преобразует мета-информацию игры, корректно восстанавливает окна предыдущих сессий (включая полноэкранный режим), и совместима с самыми новыми виртуальными машинами. Но она покамест на *Qt 3*, и в дистрибутивах только с *Qt 4* возможны проблемы, пока не появится следующее обновление.



› Вы – в лабиринте из одинаковых извилистых проходов. Нет, забудьте, вы в автомобиле…

Игр в формате *Tads* множество. Часть из них – скорее повесть, чем игра, но все они интересны, и в них есть чем заняться, кроме как стрелять/найти сокровище/повторить. Игровой процесс так же прост, как и в 8-битовую эру: читайте повествование, вводите свой вопрос, взаимодействуйте. Появился приятный бонус в виде отмены нескольких ходов, но тихая радость от текстовых приключений осталась прежней.

Гибридная клиентская система

Rootz

Версия 0.4.1 **Сайт** http://vamosproject.org/rootz

Мобильные приложения – прекрасная идея. Можно упаковать исполняемый файл со всеми его зависимостями и запускать его где угодно с флэш-карты, а это – отличный способ продемонстрировать программу или использовать ее в поездке. Представьте, что вы можете в любом месте запустить любую программу столь же легко, как у себя в локальной сети. *Rootz* дает вам такую возможность, позволяя подмонтировать ISO или директорию локально, через LAN или даже через Интернет.

Программа имеет несколько зависимостей: для всех удаленных файловых систем, с которыми намечено работать. В системах Debian, наберите

```
sudo apt-get install dchroot
sudo apt-get install libfuse2 fuse-utils
sudo modprobe fuse
```

Надо проверить, есть ли у вас SquashFS и UnionFS; если нет, сделайте следующее:

```
sudo apt-get install unionfs-modules-$(uname -r)
sudo apt-get install squashfs-modules-$(uname -r)
```

Для опциональной HTTPFS, позволяющей монтировать зеркала HTTP, проверьте, какая у вас версия *Libfuse*:

```
dpkg -l libfuse2
```

затем загрузите соответствующие версии HTTPFS с Vamos и переместите их в **/usr/bin**, и сделайте вот что:

```
sudo chmod 755 /usr/bin/httpfs
```

Теперь распакуйте пакет *Rootz*, *cd* в директорию **rootz** и введите:

```
./setup-browser
```

Если вы хотите монтировать *Rootz* стандартным образом и желаете добавить запись *Rootz* в *Fstab* (что рекомендуется), введите **./setupmount** от имени суперпользователя (root).

Документация довольно хорошая, но в переводе есть несколько недочетов. HOWTO показывает, как монтировать удаленные системы:

```
./rootz http://ubuntu-cdimage.datahop.it/releases/7.10/release/ubuntu-7.10-dvdi386.iso /mnt/ubuntu
```

Теперь вы можете запускать удаленные приложения прямо из командной строки:

```
$ sudo ./rootz-launcher whoami
[sudo] password for richard:
trying to locate whoami on ubuntu...
I: [ubuntu chroot] Running command: "which whoami"
Run whoami (ubuntu image)?
Hit anything to proceed. Use a or A to Abort
I: [ubuntu chroot] Running command: "whoami"
root
$
```

или даже из вашего браузера с помощью **rootz://ps**. Что получится, если все это проделать, показано на экранном снимке.



› Выглядит как обычный калькулятор, но в том-то и дело: *Rootz* совершенно незаметен.

Просмотрщик обложек и картинок

PictureFlow

Версия 0.1.0 **Сайт** http://pictureflow.googlecode.com

Очевидный и повсеместно признанный факт: любому компьютерщику непременно надо не отстать от друзей по части всяких красивостей и примочек. Только взгляните на все эти прошлогодние изыски вроде вращающихся кубов, прозрачных терминалов и крутящихся окон. А теперь каждому подавай iPhone, или, по крайней мере, всякие прелести блестящего GUI, которым Apple снабдил свою дорогостоящую игрушку.

PictureFlow – программа просмотра обложек, базирующаяся на *Qt*. Она демонстрирует изображения с эффектом анимации, очень похоже на *Cover Flow* от Apple, используемую в *iTunes*, и оптимизирована для работы на мобильном устройстве с Linux. Причем отсутствие самого мобильного устройства – не помеха вашему знакомству с ним прямо на рабочем столе.

Для настройки *PictureFlow* вам понадобится *Qmake*. В Kubuntu (и большинстве других дистрибутивов) он содержится в **qt3-dev-tools**. Распакуйте его и соберите:

```
tar zxvf pictureflow-0.1.0.tar.gz
cd pictureflow-0.1.0
sudo qmake-qt3 pictureflow.pro
sudo make
```

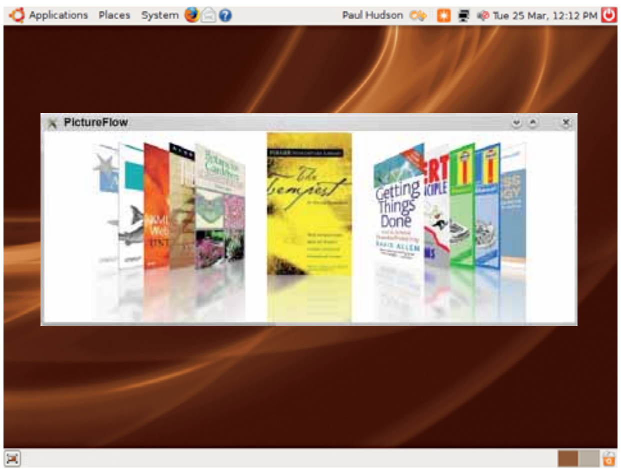
В этом первом релизе нет документации, но чтобы запустить программу, надо всего лишь указать ей на директорию с изображениями, например, так:

```
./pictureflow ~/photo/fosdem2008/
```

Вы можете поискать на YouTube видео анимационных эффектов смены изображений в действии (и загрузить его с помощью инструмента *YouTube-dl* прошлого месяца). Эффект размытия на экранном снимке – это лишь глазурь на пирожном.

PictureFlow запускается на Trolltech Greenphone и виджете Chumby Wi-Fi, и будет работать с *Qt* на других платформах, а нетребовательность к ресурсам (тут не нужно 3D-ускорения, и даже не нужны библиотеки OpenGL для эффектов перехода) означает, что он будет работать быстро на любой старой машине с установленным Linux. Ему совершенно не нужна большая производительность, он с успехом скомпилируется даже на очень маломощном процессоре.

Пока это просто развлечение, но по мере развития *PictureFlow* будет включаться во все мультимедийные приложения, слайд-шоу и портативные устройства. Можете попробовать также и *PhotoFlow* для изображений ландшафтного формата, но он пока доступен только через SVN.



› Ненужные красивости? Ну и пусть – это выглядит красиво на iPhone, значит, и ваш рабочий стол может пойти по стопам Apple.

Менеджер мобильного телефона

Wammu

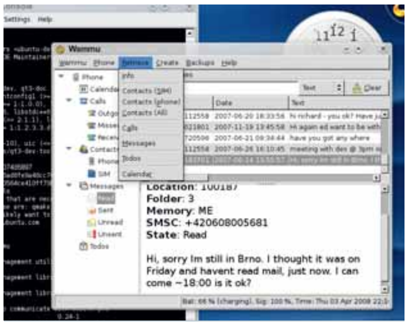
Версия 0.26 **Сайт** http://wammu.eu

Wammu позволяет управлять вашим мобильным телефоном с вашего Linux-ПК. Он написан на Python с помощью инструментария *wxPython*; в качестве движка используется *Gammu*. *Wammu* поддерживает множество телефонов – в частности, Sony-Ericsson, Nokia, Motorola и большинство устройств Symbian; все они (вместе с доступными функциями) перечислены в телефонной базе данных *Gammu*, на которую можно выйти через меню Справка.

Большинство дистрибутивов предоставляет *Wammu* в виде готового пакета, а Deb'ы, RPM и исходный код можно найти на сайте. Удостоверьтесь, что ваш телефон обнаруживается по Bluetooth, возьмите дата-кабель или разберитесь с инфракрасным портом на ноутбуке, а затем подключитесь к телефону с помощью мастера.

Чаще всего *Wammu* используется для загрузки контактов, звонков, текстовых сообщений и пометок в календаре. Любое приложение *Gammu* (или *Gammu+*) умеет сохранять загруженные данные в виде файла; можно экспортировать сообщения как электронную почту. И это вовсе не односторонне движение: контакты, календарные пометки и списки срочных дел могут создаваться на вашем ПК и загружаться в телефон. Вы также можете составлять SMS-сообщения, не тратя нервы на T9 и неудобные мелкие кнопки.

В некоторых телефонах есть возможность загрузки других файлов, в том числе рингтонов, и каждый релиз усовершенствует поддержку мультимедиа в составлении SMS. Загляните в подробный перечень функций, поддерживаемых для вашего телефона, в специальном списке.



› **Загружайте, редактируйте и сохраняйте свои сообщения, контакты и прочую телефонную информацию.**

ПО для блогов

WordPress

Версия 2.5 **Сайт** www.wordpress.org

LINUX FORMAT HotPicks *повторный визит*

Иногда создается впечатление, что если у вас нет блога, то вы – представитель очень скудного и молчаливого меньшинства. Но программы для блогов занимаются вовсе не одним выпусканием паров. Например, *WordPress* не только популярен для блогов, но и создает отличную основу для простой системы управления контентом.

Со времени нашего прошлого обзора в LXF65, *WordPress* ушел далеко вперед. В последнюю версию добавлены загрузка множественных файлов, настраиваемая информационная панель, шифрование паролей и cookie, библиотека мультимедиа, редактор WYSIWYG, который не путается с кодом, защита от параллельного редактирования постов, полноэкранный режим набора текста и функция поиска по постам и страницам.

Wordpress.org снабжен удобным руководством по обновлению и установке с нуля. Требования – PHP 4.3 и выше и *MySQL 4.0* – весьма консервативны и доступны большинству недорогих или бесплатных хостинг-провайдеров. Для абсолютных новичков есть даже книга под названием «*WordPress для "чайников"*», описывающая версию 2.3.

WordPress на диво популярен, что привело к появлению настоящей сокровищницы тем и модулей расширения. Первые вполне достойны исследования, а со вторыми надо обращаться с осторожностью: в любом проекте PHP код, созданный третьей стороной, всегда создает риск с точки зрения безопасности. Но у тех, кто всего лишь размещает с помощью *WordPress* интранет-блог или CMS, нет повода для беспокойства. LXF

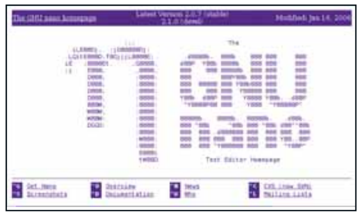


› **Вам есть что сказать? Поместите это в блог; а если это ролик на YouTube — встройте.**

Также вышли

Новое и обновленное ПО, также заслуживающее внимания…

GNU Nano 2.1.0 (unstable)
Внушительный клон текстового редактора Pico позволяет менять привязки клавиш.
www.nano-editor.org



Physics-Live CD 1.0
Загружаемые физические модели на основе Slax; добавлены математика и химия.
www.physics-live.org

Cruiser 0.1.0
Браузер системного лотка для файлов и новостных лент.
http://cruiser.sourceforge.net

ADempiere 3.4.0
ERP, рынок, и вообще – решение для бизнеса.
www.adempiere.com

Schoorbs 1.0 Beta 2
Бронирование мест и ресурсов с web-интерфейсом 2.0.
http://schoorbs.xhochy.com

Cfunge 0.2.0
Маленький, быстрый интерпретатор Befunge-98 на С. Вот что я называю потехой!
http://freshmeat.net/redir/cfunge/73691/url_homepage/cfunge

Poker-network 1.3.0
Сервер и библиотека клиента для покера: обогатитесь с помощью FOSS и рудиментарной математики.
http://gna.org/projects/pokersource

FLTK 1.1.8
Лаконичный кросс-платформенный GUI-инструментарий для C++ пополнился новой темой в стиле GTK, улучшенной совместимостью с Glut 4 и альфа-смешением для изображений-карт.
www.fltk.org

BeeDiff 1.6
Графически сравнивает текстовые файлы.
www.beesoft.org/beediff.html

Vlock 2.2
Блокирует виртуальные консоли на машинах с открытым доступом.
http://cthulhu.c3d2.de/~toidinamai/vlock/vlock.html

Mahara 1.0.0
Портфолио, блог, программа создания резюме и социальной сети для организации обучения и развития пользователей на протяжении всей жизни, от Новозеландской академии.
www.mahara.org

LXF DVD107

Сметите все с рабочего стола и попробуйте что-нибудь новенькое!



Хотите попробовать новый дистрибутив? Тогда вы обратились по адресу! На DVD этого месяца – две полноценных версии Linux: Ubuntu 8.04 (традиционный «большой» релиз от команды **LXF**) и Mandriva 2008 Spring. Каждая из них хороша по-своему: обзоры, опубликованные в этом номере журнала и текст на последующих четырех страницах, подскажут, что лучше всего подойдет именно вам!.

Майк Сондерс
Новостной редактор
mike.saunders@futurenet.co.uk

Дистрибутив Linux

Ubuntu 8.04

Вы, наверное, уже читали о новых функциях Ubuntu 8.04 (если нет, откройте стр. 12), поэтому мы не будем подробно описывать их здесь. Достаточно сказать, что это крупный релиз от Canonical, второй по счету с долгосрочной поддержкой (Long Term Support, LTS), что означает три года обновлений для настольных и пять лет – для серверных приложений. Если вы до сих пор работаете в Ubuntu 6.06, не используя промежуточных релизов в ожидании версии LTS, сейчас самое время перейти к 8.04.

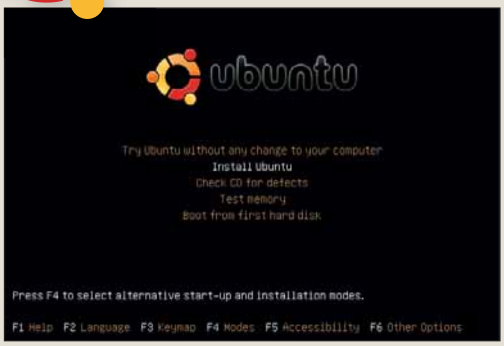
На вашем **LXFDVD** вы найдете мощную версию Ubuntu с большим количеством дополнительных пакетов. Вы обожаете Ubuntu, но предпочитаете KDE? Или вас потряс *Xfce*? В любом случае, можете запустить свое любимое окружение сразу после установки с нашего DVD – здесь собрано воедино все лучшее от Ubuntu, Kubuntu и Xubuntu, и можно сменить рабочий стол прямо из окна входа в систему. Мы также включили сюда дополнительные программы, не поставляемые с Ubuntu по умолчанию, такие как *Thunderbird и Scribus*, а также инструменты разработки и библиотеки. Если у вас телефонное интернет-соединение (или его нет вообще), мы надеемся, что эти дополнительные программы вам пригодятся. И даже если у вас скоростной ADSL-канал, они помогут вам сэкономить время и силы.

Как и в предыдущих версиях, Ubuntu работает в режиме Live, и вы сможете познакомиться с дистрибутивом, не инсталлируя его на жесткий диск. Но если вы пожелаете установить его в качестве постоянной системы, следуйте приведенным ниже инструкциям. Минимальные системные требования для нормальной работы Ubuntu 8.04:

- 800-МГц Intel-совместимый CPU
- 384 МБ ОЗУ
- 10 ГБ места на жестком диске.

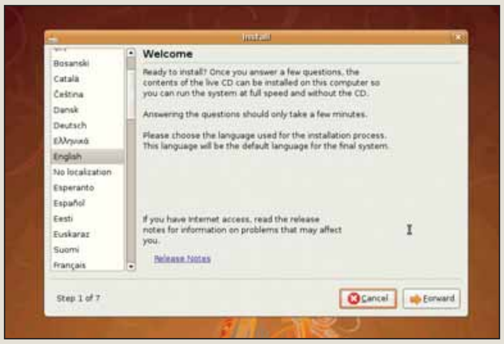
Обратите внимание, что на нашем DVD – 32-битная версия, но она с успехом будет работать на 64-битной машине. Вообще-то многие

Шаг за шагом: Устанавливаем Ubuntu Linux 8.04



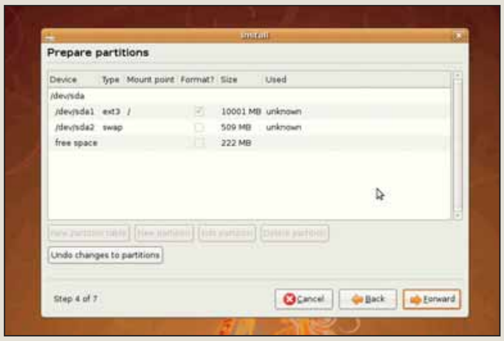
1 Загрузка

Загрузите компьютер с **LXFDVD**, и когда появится этот экран, используйте кнопки со стрелками и Enter, чтобы выбрать Install Ubuntu.



2 Запуск

Через пару минут появится экран программы установки. Прочитайте текст и нажмите Forward.



5 Разбивка вручную

Если вы решили разбить диск вручную, создайте корневой раздел (/) в формате ext3 размером не менее 10 ГБ и раздел подкачки (виртуальной памяти) размером 512 МБ.



6 Создание учетной записи

Когда появится этот экран, создайте учетную запись. Не забывайте свой пароль, и помните, что регистр имеет значение!

владельцы 64-битных ПК используют 32-битные дистрибутивы, поскольку они гарантируют совместимость с 32-битными программами. Но если вы хотите максимизировать производительность своей машины, можете прожечь ISO-образ из раздела Дистрибутивы нашего DVD и загрузиться с него (это обычная версия Ubuntu, без дополнительных программ).

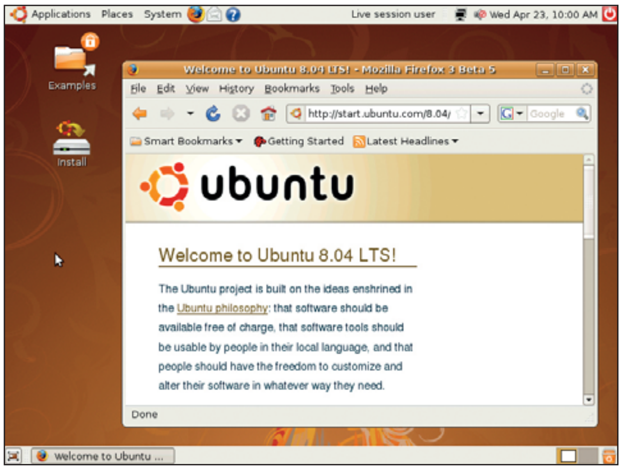
Новая жизнь старых ПК

Для тех, кто хочет установить Ubuntu на старую машину (минимум со 128 МБ ОЗУ), мы предлагаем «альтернативный» ISO-образ 8.04. Он сразу загружает текстовую программу установки, и хотя с ней не так просто работать, как с обычной версией, она съедает значительно меньше памяти и позволяет выполнить автоматическую установку на других компьютерах. Если вы инсталлируете Ubuntu 8.04 на старый ПК, вам придется отказаться от Gnome и KDE в пользу рабочих столов полегче (типа *IceWM* или *Fluxbox*). Воспользуйтесь альтернативным ISO, если хотите установить обычную версию Ubuntu, без всяких дополнительных программ.

Программа установки Ubuntu позволяет изменить размер разделов на жестком диске. Если вы отводите под Linux весь винчестер, вас не должно это волновать, но если вам нужна двойная загрузка с Windows, сперва надо будет освободить некоторое пространство на ее разделе. Убедитесь, что у вас есть не менее 10 ГБ свободного места, а потом запустите проверку диска и Дефрагментатор в Windows. Не забудьте сделать резервные копии важных данных перед началом установки Ubuntu! Инсталлятор позволит вам сжать раздел Windows и выделить место для нового раздела Linux.

Hardy Heron – первый релиз Ubuntu, включивший *Wubi*, программу, позволяющую установить и запустить дистрибутив на Windows-машине без переразбивки жесткого диска. Такая установка немного медленнее обычной, и в ней нет некоторых функций – например, режима ожидания; но все же это – отличный способ познакомить ваших родных и друзей с миром Linux. В Windows запустите **umenu.exe** с DVD, чтобы активировать *Wubi* – помните, что это новая функция в дистрибутиве, и поэтому нелишним будет сделать резервные копии важных данных!

Если при установке или во время работы с Ubuntu возникнут проблемы, зайдите на www.linuxforum.ru: здесь вы встретите множество пользователей Ubuntu, готовых помочь. Или посетите англоязычные www.ubuntuforums.org, там есть множество разделов, от поддержки оборудования до безопасности. Если же вы хотите нести информацию в массы или добыть диски для друзей или коллег, зайдите на http://shipit.ubuntu.com, сервис бесплатного распространения дисков Canonical. Будет много желающих заиметь самый свежий релиз, так что возможны некоторые задержки в доставке, но вы получите свой диск Ubuntu в красивой обложке, способствующей делу распространения Linux. »



› Ubuntu 8.04 (он же Hardy Heron) включает изумительно быстрый web-браузер *Firefox 3.0*, впечатляюще солидный, несмотря на статус бета-версии.

Пользователи Xfce

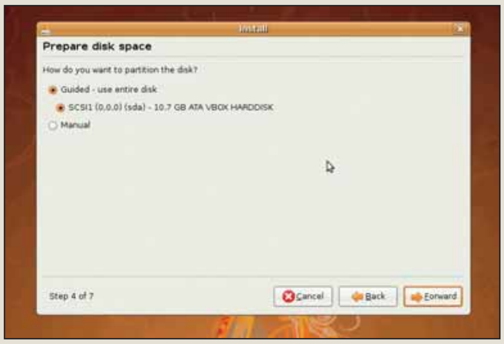
По техническим причинам, *Xfce* нельзя запустить через менеджер *GDM*. Перед вами окажется пустой экран, и придется нажать Ctrl+Alt+Backspace, чтобы выйти из него. Чтобы исправить это, зайдите в Gnome, откройте терминал и введите следующее:

```
sudo gedit /usr/share/xsessions/xfce4.desktop
```

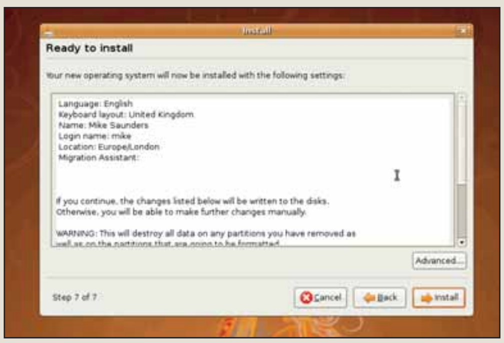
Измените строку Exec= внизу на Exec=xfce4-session, сохраните файл и можете выходить. Теперь будет запускаться нормально.



3 Регион
Выберите свой регион на карте мира, затем укажите раскладку клавиатуры и нажмите Forward [Далее].



4 Разделы
Теперь можно заняться разделами жесткого диска: измените размер уже существующих или отведите под Ubuntu весь диск целиком.



7 Подтверждение
Программа установки спросит, хотите ли вы продолжить установку. Если вы решили что-то поменять, нажмите Back [Назад].



8 Ждите
Ubuntu Linux начнет копирование на жесткий диск (это может занять 10–30 минут). По окончании этого процесса ваш компьютер перезагрузится, так что извлеките DVD и наслаждайтесь!

Дистрибутив Linux

Mandriva Linux 2008

Полное название данного релиза – Mandriva Linux Free 2008 Spring. Не вдруг и упомнишь, поэтому мы считаем нужным объяснить: это – свободный дистрибутив Mandriva Linux (www.mandriva.com); 2008 – номер базовой версии, а Spring – ревизия. Вы, возможно, помните, что Mandriva 2008 была на DVD LXF102; релиз Spring освежен новейшими пакетами и дополнительными функциями.

Mandriva стабильно удерживает позиции одного из лучших дистрибутивов для начинающих, сочетая дружелюбный к пользователю процесс установки с богатым выбором инструментов настройки, так что вам не грозит вылазка на территорию командной строки. Более подробную информацию по этому дистрибутиву вы найдете на стр. 16; также посмотрите файл **i586/index.htm** на второй стороне LXFDVD.

По вашим многочисленным просьбам мы публикуем Free-редакцию Mandriva Linux 2008 Spring: ее нельзя запустить прямо с DVD (как Mandriva One), но зато она содержит максимальное количество пакетов свободных приложений. Чтобы начать установку, просто загрузитесь со второй стороны LXFDVD и выберите пункт Install Mandriva Linux 2008 Spring. Если что-то пойдет не так, перезагрузитесь и нажмите F3, а затем попробуйте поэкспериментировать с параметрами ядра (меню вызывается клавишей F6).

Что вам понадобится

Основные системные требования для Mandriva:

- Процессор Intel или AMD, 32- или 64-битный;
- 256 МБ ОЗУ;
- 3 ГБ места на жестком диске.

Для хорошей работы рекомендуем 512 МБ ОЗУ и не менее 10 ГБ свободного пространства на винчестере.

Славящийся своей дружелюбностью инсталлятор Mandriva Linux проведет вас через все этапы установки: от выбора языка до настроек сети. После нескольких предварительных вопросов, вам будет предложено разбить жесткий диск на разделы. Если вы работаете в Windows, вам понадобится освободить место на винчестере (рекомендуем не менее 5 ГБ). Программа установки может сжать раздел Windows, высвободив место для Linux. Если же вы готовы отвести под Mandriva весь диск целиком, то на стадии создания разделов выберите Erase And Use Entire Disk [Очистить и использовать весь диск]. Продвинутые пользователи, как и всегда, могут прибегнуть к ручной разметке – но в любом случае, рекомендуем вам сначала создать резервную копию ценных данных.

Вторая важная фаза – выбор пакетов для установки. И опять же, вы можете позволить инсталлятору выполнить большую часть работы за вас, а можете любовно отбирать пакеты по одному (хотя программа все равно позаботится обо всех необходимых зависимостях). Не забудьте также указать рабочий стол: это вопрос вкуса, но Mandriva традиционно больше тяготеет к KDE. Затем начнется копирование файлов на жесткий диск и будет произведена окончательная настройка. Когда вам предложат перезагрузить компьютер, не забудьте вынуть из привода LXFDVD или же выбрать загрузку с жесткого диска в настройках BIOS.

> «Mandriva удерживает позиции одного из лучших вариантов для начинающих.»



› Чтобы настроить Mandriva Linux (например, установить новое ПО или оборудование), нажмите на значок Configure Your Computer справа от кнопки Menu.



› Mandriva One работает прямо с DVD и ставится в несколько щелчков, но во Free-редакции больше пакетов: мы выбрали второе.

Техническая поддержка

Если вдруг вы увязнете при установке или во время работы с Mandriva, выходите в Сеть и отправляйтесь на *Линуксфорум*, расположенный по адресу www.linuxforum.ru. Там вы найдете других пользователей, которые, возможно, сталкивались с теми же проблемами, что и вы – объясните им, что случилось, и кто-нибудь обязательно поможет вам советами и подсказками.

Однако есть основания полагать, что все пройдет успешно, так что радуйтесь своей новой установке Linux!

Очень ВАЖНО!

Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим:

Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru для получения содействия.

ВНИМАТЕЛЬНО ПРОЧТИТЕ ЭТО ПЕРЕД ТЕМ, КАК ИСПОЛЬЗОВАТЬ DVD-ДИСК !

Информация о диске

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

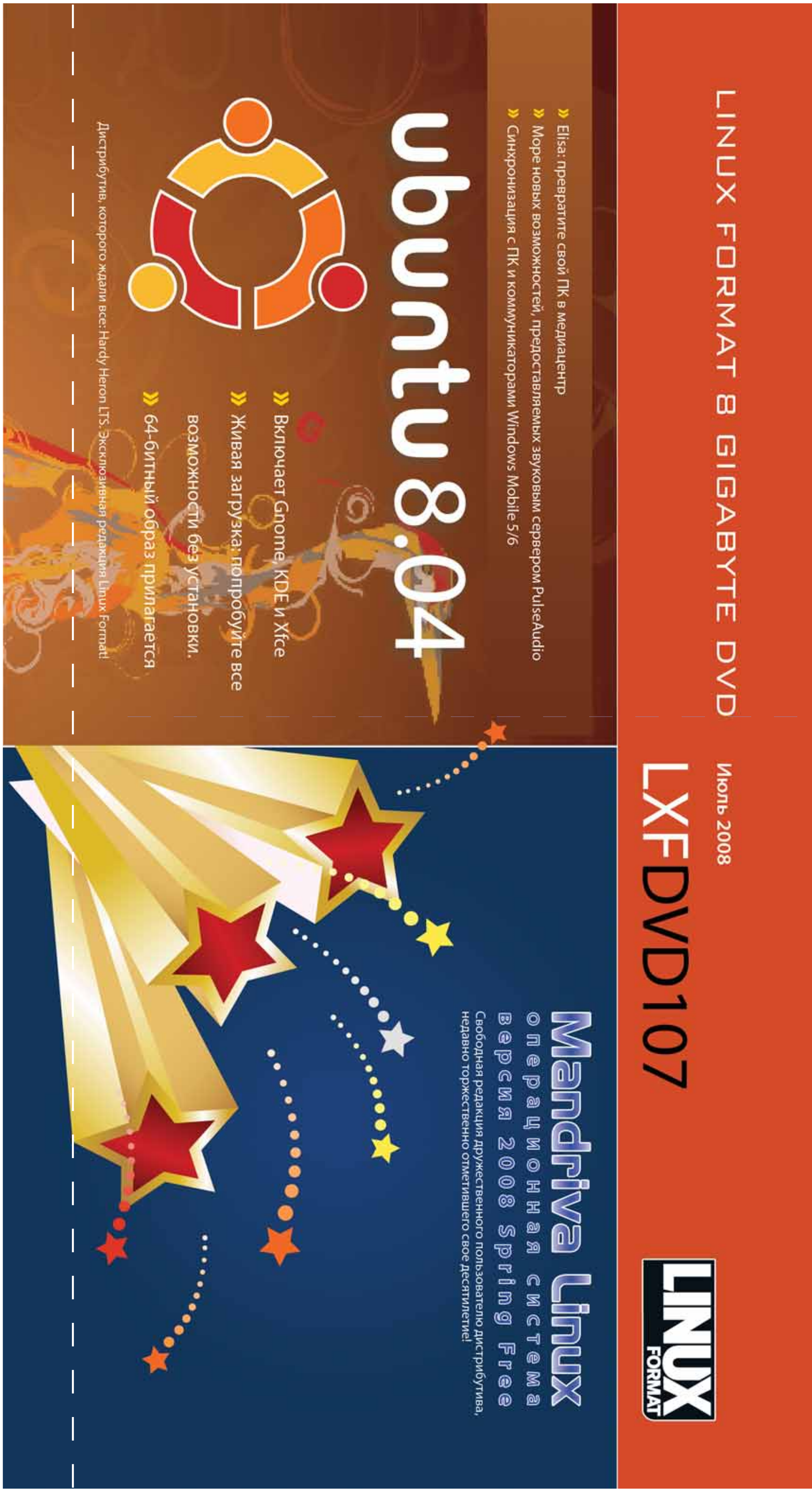
имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru



DVD

Содержание **LINUX FORMAT DVD Июль 2008**

LINUX FORMAT

Сторона 1

Рабочий стол
BasKet - программа для создания корзин
Compiz - оконный менеджер
GCompris - комплект обучающих программ
GIMP - растровый редактор
Inkscap - открытый векторный редактор
Kino - DV-редактор
KnowIt - инструмент для управления заметками
MythTV - универсальный мультимедиа-комбайн
NoteCase - менеджер заметок
Pondus - программа для слежения за вашим весом
Project Hamster - инструмент для учёта времени
Tomboy - программа для создания заметок в Wikiподобном стиле
TuxCards - создание заметок
Woindow Maker - оконный менеджер в стиле NeXT
Zim - WYSIWYG-редактор

Разработка
Arduino_IDE - среда разработки для микроконтроллеров
Cream - вариация Vim
Geany - среда разработки
Pygame - набор модулей Python для создания игр

Дистрибутивы
Ubuntu - последняя версия популярнейшего дистрибутива
Zenwalk - легковесный дистрибутив на основе Slackware

Игры
Atomic_Tanks - двумерные танковые сражения
House Of Mirrors - логическая игра
Stormbaan Coureur - новая версия Sturmbahnfahrer

Справка
Ответы на часто задаваемые вопросы
Новичку в Linux
RUTE - книга по администрированию Linux-систем

Hot Picks
Chandler_Desktop - программа для создания списка задач из входящих
PictureFlow - виджет Qt/Qtopia для отображения картинок
QTads - интерпретатор текстовых приключенческих игр
Risk - Java-версия классической игры Risk
rootz - система распространения программ
SoundConverter - конвертера аудиоформатов
Super Grub Disk - загрузочный образ для восстановления GRUB
Wammu - менеджер мобильного телефона

Незаменимое
Alexandria - программа для каталогизации книжной коллекции
Bluefish - мощный HTML/XHTML/XML-редактор
CeeMedia - программа для каталогизации вашей коллекции фильмов
Cuyo - вариация тетриса
Dia - программа для создания диаграмм
Hugin - приложение для составления панорам
Jokosher - аудиоредактор
GNU Octave - высокоуровневый язык численных вычислений
Qtpfsgui - графический интерфейс для создания HDR
Salasaga - среда разработки видеокурсов

Интернет
Claws Mail - почтовый клиент
Firefox - популярный веб-браузер
gnubiff - программа для уведомления о новых сообщениях электронной почты
KuFTP - графический FTP-клиент
Opera - набор приложений для сети Интернет
SkipStone - веб-браузер на Gtk+

Офис
OpenOffice.org - русскоязычная сборка офисного пакета

Система
Evilvte - эмулятор терминала
Rxlogd - коллектор syslog
Schily - пакет утилит командной строки

Поставляется вместе с журналом LINUXFORMAT номер 7 (107) Июль 2008

Сторона 2

Mandriva 2008 Spring Free

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен ООО «Фортмедиа», Россия, Санкт-Петербург, 196006 ул. Цветочная д. 7, тел. +7 (812) 388-8290. Лицензия МПТР России ВАФ № 77-225

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

Новое ПО

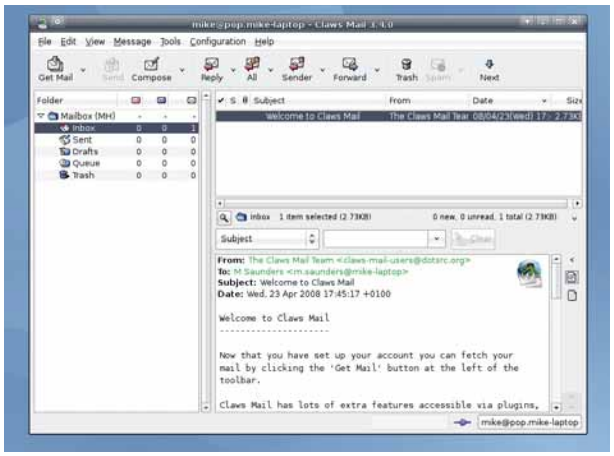
Рабочий стол и Интернет

Вместе с дистрибутивами на нашем DVD имеются самые свежие, самые лучшие приложения. В погоне за новыми релизами дистрибутивов разработчики Open Source стремятся поскорее выпускать новые версии программ, в надежде, что их детище успеет попасть в репозитории. Так что в этом месяце работа прямо кипела, и мы включили для вашего ознакомления несколько отличных приложений. В разделе Рабочий стол предлагаем вам две программы, принадлежащие к весьма обширной категории «отслеживания личных данных».

Project Hamster – это инструмент планирования времени для Gnome, который подводит итоги вашей деятельности за день. При каждой смене рода деятельности (например, занимаетесь вы со счетами, или возитесь с файлом **xorg.conf**), вы вводите свой статус в окне *Project Hamster*. А если у вас весь день не было доступа к компьютеру, можете указать эти данные, когда к нему вернетесь. Спустя некоторое время вы сможете сгенерировать статистику своей повседневной деятельности. Она может быть хорошей («Ух ты, я целых три часа посвятил учебе»), а может и плохой («Боже, я целых шесть часов смотрел видео на YouTube»), но, в любом случае, она поможет вам лучше организовать свою жизнь.

Будьте в форме

Также в разделе **Рабочий стол** вы найдете *Pondus*, программу персонального управления весом, написанную на Python: вы ежедневно вводите данные о своем весе, а она, накопив нужную статистику, составляет график. Программа может использовать данные в килограммах или фунтах и импортировать/экспортировать их в формате CSV.



› Сыты по горло раздутыми почтовыми клиентами? В *Claws Mail* множество полезных функций, но он не сжирает всю наличную память.

Вы можете задать вес, к которому стремитесь, но, как всегда уточняют в рекламе антиникотиновой жевательной резинки, тут нужна сила воли. А если вы все же хотите избавиться от нескольких фунтов к лету, *Pondus* в содружестве с *Project Hamster* смогут вам помочь.

В разделе Интернет нашего DVD вас ждут новинки сезона: *Firefox 3.0* и *Opera 9.5*. Уже несколько лет *Firefox* ругают за неумеренное потребление ОЗУ и медлительность; в версии 3.0 эти проблемы исчезают. Исправлены утечки памяти, и хотя предстоит еще много работы, этот браузер сможет вновь отвоевать тех пользователей, что предпочли перейти на *Konqueror* или *Opera*.

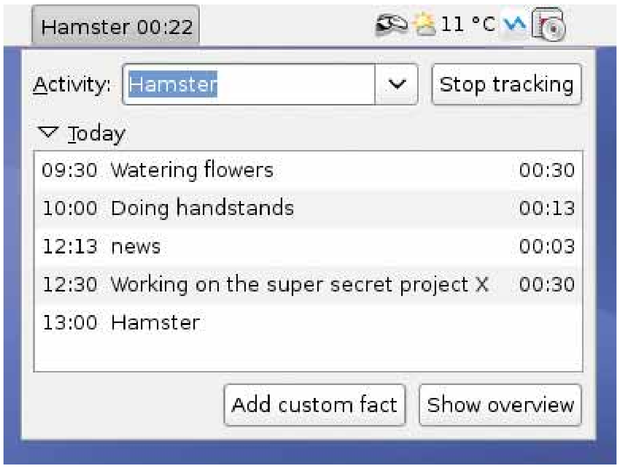
Новинка для сетежителей

Скопируйте архив *Firefox* 3.0 с компакт-диска в свой домашний каталог, распакуйте командой **tar xjvf firefox-3.0.tar.bz2**, а затем введите следующую команду для запуска браузера:

```
firefox/firefox
```

Теперь можете приступать к просмотру любимых сайтов, в особенности тех, где много графики и имеется JavaScript: сравните, насколько увеличилась скорость. Если вы хотите постоянно использовать этот релиз, создайте на рабочем столе ярлык, ведущий на скрипт **firefox** в директории **~/firefox**.

На LXFDVD этого месяца вы также найдете новый релиз *Claws Mail* – 3.4. Если раньше вы никогда не пользовались этим почтовым клиентом, настоятельно рекомендуем попробовать: он исключительно быстр и при этом имеет множество полезных функций, например, множественные учетные записи, обработка сообщений, фильтры, поддержка новостей NNTP, цветные метки и заголовки, определяемые пользователем. Для постоянных пользователей *Claws* сообщаем, что среди новых функций релиза 3.4 – поддержка тем PNG, ограничение больших сообщений (вам больше не придется просматривать все 5 МБ текстового файла сразу) и опция поиска в настройках фильтра.



› Наведите порядок в своей жизни с помощью *Project Hamster* и отделайтесь от всех бессмысленных занятий, отнимающих ваше время.

Другое ПО

Три лучших игры

Установив новый дистрибутив и познакомившись с самыми свежими приложениями, вы, без сомнения, захотите расслабиться с помощью качественных программ из раздела Игры. Берите кофе/пиво, и – вперед!

Во-первых, у нас есть новый, 2.9 релиз *Atomic Tanks*, комичной игры в стиле Worms, где вы сражаетесь с другими игроками и приобретаете оружие для следующих битв. Концепция проста: вы управляете танками на двумерном игровом поле, палите в воздух из всяческого оружия, и надеетесь, что эти залпы угодят в противников. Получаться будет не всегда, поскольку танки могут перемещаться, так что удовольствие от прямого попадания гарантировано огромное!

Далее – забавный паззл под названием *House of Mirrors* [Зеркальный Дом]. Он требует наличия языка программирования Scala (который имеется в каталоге зависимостей, а может, и в репозитории вашего дистрибутива). Установив Scala, извлеките архив *Mirrors*, перейдите в полученную директорию и введите **./run.sh** для запуска игры. (Вы найдете информацию по распаковыванию программ с помощью командной строки в разделе **Справка/Новичку** в Linux нашего DVD). Игра очаровательно необычна: у вас есть источник света и зеркала, и ваша задача – направить луч на определенное число детекторов.



› *Atomic Tanks*: классические Worms с уклоном в космическую тематику.

И, наконец, у нас есть *Stormbaan Coureur*, ранее известный, как *Sturmbahnfahrer*. Это – гонки с акцентом на физике и управлении: они не похожи на аркаду, как *Ridge Racer*. Вы едете по маршруту с препятствиями, пытаясь не упасть с платформ и эстакад. Но, несмотря на серьезное внимание, уделенное точности действий с физической точки зрения, игра доставит вам большое удовольствие, когда вы будете маневрировать на машине и врезаться в разные предметы. Чтобы запустить *Stormbaan Coureur*, вам понадобятся библиотеки *PLIB* и *ODE*; если в репозиториях вашего дистрибутива они отсутствуют, ищите исходные тексты в подкаталоге зависимостей. LXF

LINUX FORMAT
Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 6000 экз.

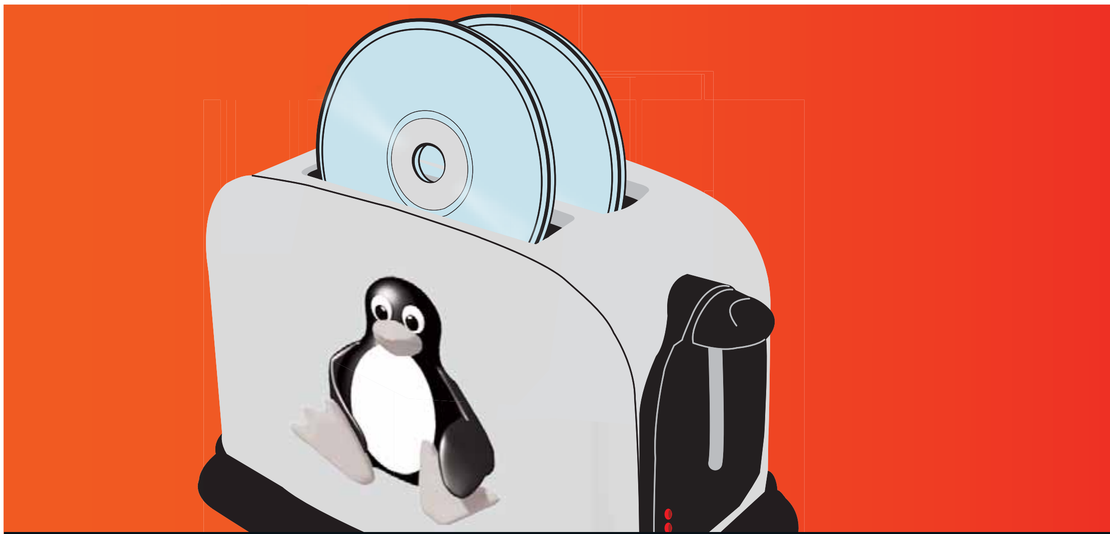
РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
Главный редактор
Валентин Синицын info@linuxformat.ru
Выпускающий редактор
Родион Водейко
Литературные редакторы
Елена Толстякова
Переводчики
Илья Аввакумов, Александр Бикмеев, Светлана Кривошеина, Александр Казанцев, Алексей Опарин, Валентин Развозжаев
Редактор диска
Александр Кузьменков
Допечатная подготовка
Мария Пучкова, Родион Водейко, Семен Фрумкин
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Генеральный директор
Павел Фролов
Заместитель генерального директора
Софья Виниченко
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____
Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Пол Хадсон [Paul Hudson] paul.hudson@futurenet.co.uk
Редактор новостей Майк Сондерс [Mike Saunders] mike.saunders@futurenet.co.uk
Редактор обзоров Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Литературные редакторы Эндрью Грегори [Andrew Gregory] agregory@futurenet.co.uk, Мэтт Нейлон [Matt Nailon] mnailon@futurenet.co.uk

Подготовка материалов
Ладислав Боднар [Ladislav Bodnar], Нейл Ботвик [Neil Bothwick], Джон Брэндон [John Brandon], д-р Крис Браун [Dr. Chris Brown], Энди Ченнел [Andy Channelle], Майкл Дж. Хэммел [Michael J. Hammel], Ричард Хилсли [Richard Hillesley], Энди Хадсон [Andy Hudson], Ричард Смедли [Richard Smedley], Ник Вейч [Nick Veitch], Евгений Балдин, Александр Бикмеев, Андрей Боровский, Дмитрий Кирсанов, Алексей Маслий, Станислав Механошин, Игорь Новиков, Галина Пожарина, Олег Попов, Андрей Прахов, Алексей Федорчук, Илья Шпаньков, Никита Шультайс
Художественные ассистенты: Зигги Бейкер [Ziggi Baker], Нейл Фристоун [Neil Freestone], Клэр Герри [Claire Gerrey], Фил Хэйкрэфт [Phil Haycraft]
Ассистент по выпуску: Сара Эдвардс [Sarah Edwardes]
Иллюстрации: Крис Винн [Chris Winn], iStock Photo

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург (редакция): Лиговский пр., 50, к.15, тел. +7 (812) 640-49-90
Представительство в Москве: ул. Энергетическая, д.14, корпус 5, стр. 1
По вопросам сотрудничества, партнерства, оптовых закупок: +7 (495) 799-18-63
Дирекция московского офиса: +7 (495) 136-88-45. E-mail: moscow@linuxcenter.ru



в августе

Linux на чем угодно

Готовы к фестивалю инсталляций? Заприте свои гаджеты, пока им еще ничего не угрожает – мы намерены показать, как водрузить туда Linux.

НОВИНКА: Ruby on Rails

Учитесь программировать по правилам Web 2.0 в нашей новой серии учебников.

Заставьте ALSA работать

Все пользуются звуковой картой, но не у каждого она работает безупречно. Это – ваш шанс перейти на светлую сторону.

Нелюди говорят!

Научите компьютер читать вам новости вслух с помощью Festival, RSS и хакерской магии.

Содержание последующих выпусков может меняться без уведомления